

The Definitive Certified ScrumMaster® (CSM) Workbook

Daniel James Gullo

CEC, CST

Copyright © 2012-2025

Daniel James Gullo
CEO, Principal

Apple Brook Consulting®
PO Box 369
Cochranville, PA 19330-0369

daniel.gullo@apple-brook.com

Disclaimer
Every effort possible has been made to provide proper attribution for the sources in this training handbook. Any missed attribution or misattribution is unintentional. Please contact us immediately, and we will revise accordingly.



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Table of Contents

Build Your Own Scrum 5
Scrum Overview 7
The New New Product Development Game 8
What Does SCRUM Stand For? 9
Scrum Framework 10
Define empiricism and list the three empirical Scrum pillars 11
List the Five Scrum Values 12
Significant Organizational Change 13
List the benefits of an iterative and incremental approach 14
Scrum Roles 15
The Scrum Team 17
CSM-1.1 Describe the responsibilities and accountabilities of the Scrum Team 18
Product Owner 21
CSM-1.4 Describe the responsibilities and accountabilities of the Product Owner 22
CSM-1.5 Discuss at least two reasons why the Product Owner is a single person and neither a group nor a committee 24
CSM-1.6 Discuss how and why the Product Owner maintains authority over the Product Backlog 25
while working collaboratively with Developers and stakeholders 25
CSM-3.9 Discuss why Scrum does not have a project manager 26
Developers 27
CSM-1.3 Describe the responsibilities and accountabilities of the Developers 28
Scrum Master 31
CSM-1.2 Describe the responsibilities and accountabilities of the Scrum Master 32
Scrum Events and Activities 44
CSM-1.7 Identify at least one example of how a Scrum Team could inspect and adapt to increase transparency at each of the Scrum events 46
The Sprint 47
CSM-1.14 Explain under what conditions a Sprint could be terminated prematurely 51
Sprint Planning 54
Daily Scrum 60
CSM-1.12 Explain how Developers conduct a Daily Scrum 61
CSM-1.13 Discuss at least three ways the Daily Scrum differs from a status meeting and why the various constraints exist to support the Developers 61
Product Backlog Refinement 65
Sprint Review 68
Sprint Retrospective 74
CSM-1.11 Describe at least three possible effects of skipping the Sprint Retrospective 77
Scrum Artifacts 79
Product Backlog 80
Sprint Backlog 89
The Increment 94
Definition of Done 98
CSM-1.15 Explain at least three advantages of a strong Definition of Done 99
CSM-1.16 Outline at least one way to create a Definition of Done 99
CSM-3.2 Explain the impact of accumulating technical debt 100
Agile Facilitation 104
CSM-2.1 Describe at least three situations in which the Scrum Master could serve the needs of the Scrum Team 105
CSM-2.2 Demonstrate at least three techniques for facilitating group decision making 106
Agile Coaching 108
CSM-2.3 Discuss how facilitating, teaching, mentoring, and coaching are different 109
Service to the Scrum Team 111
CSM-3.1 Describe three scenarios where the Scrum Master acts as a leader for the Scrum Team 114
CSM-3.7 Apply at least one technique that could help resolve an impediment CSM-3.6 Discuss at least two ways that the Scrum Master assists the Scrum Team with impediments 115
CSM-3.3 List at least three development practices that could help their Scrum Team 119

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Service to the Product Owner **124**
CSM-3.4 Explain at least three ways the Scrum Master could support the Product Owner *125*
Service to the Organization **130**
CSM-3.8 Summarize at least one organizational design change caused by adopting Scrum *131*
Organizational Change Models *135*
CSM-3.5 Describe at least three organizational impediments that can affect Scrum Teams *136*
Scrum Simulation **141**
CSM-1.8 Perform a Sprint Planning *141*
CSM-1.9 Perform a Sprint Review *141*
CSM-1.10 Perform a Sprint Retrospective *141*
CSM-2.2 Demonstrate at least three techniques for facilitating group decision making *141*
Afterword **143**
Thank You!! *144*
*For the **one person** who didn't get the joke…* *147*
What's Next? **149**
Appendix **151**
The Scrum Guide 152
Professional Guessing 166
Bibliography **183**
References and Further Reading *184*
Culturama *188*

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Build Your Own Scrum

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

ACTIVITY

The class will be divided up into teams.

Each team will have their own breakout room and Mural space to work on. The goal for your team is to create a visual, flow diagram that teaches and explains the Scrum framework according to what the Scrum Guide says.

You may use the provided elements in the Mural space or you can use your own elements which you borrow from the Internet and upload from your own machine(s).

You will have **30 minutes** for this exercise. At the end, you and your team should be prepared to present and explain your diagram.

Please ensure that the content you select is PG rated.



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum Overview

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

The New New Product Development Game

Authors
Hirotaka Takeuchi
Ikujiro Nonaka

Publication
Harvard Business Review, January 1986

Summary
This article was a key inspiration for Schwaber and Sutherland in their creation of Scrum

The typical waterfall approach to software development, is compared to a relay race, where there is little collaboration between the team members.

The sign-offs in the phases of traditional software development are akin to passing of the baton in a relay race.

Once each group has done its part, the success is no longer their “problem”.

Instead, following a more rugby related approach where the teams are working closely together and passing the “ball” back and forth to accomplish their goals is a more preferred approach.

Collaboration happens, and the teams are more self-organizing so that they can figure out whatever needs to be done to achieve success.

The article includes numerous case studies from development projects across various industries.

Several graphics describe what we now call “Rational Unified Process” with the phases of development overlapping instead of being discrete and gated with formal sign-offs.



Hirotaka Takeuchi



Ikujiro Nonaka

Source: https://hbr.org/1986/01/the-new-new-product-development-game

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

What Does SCRUM Stand For?

In reality, Scrum doesn't stand for anything. It's not an acronym at all. It refers to the formation in rugby that occurs when the ball is brought back into play. It's the coming together of teams in order to move forward.



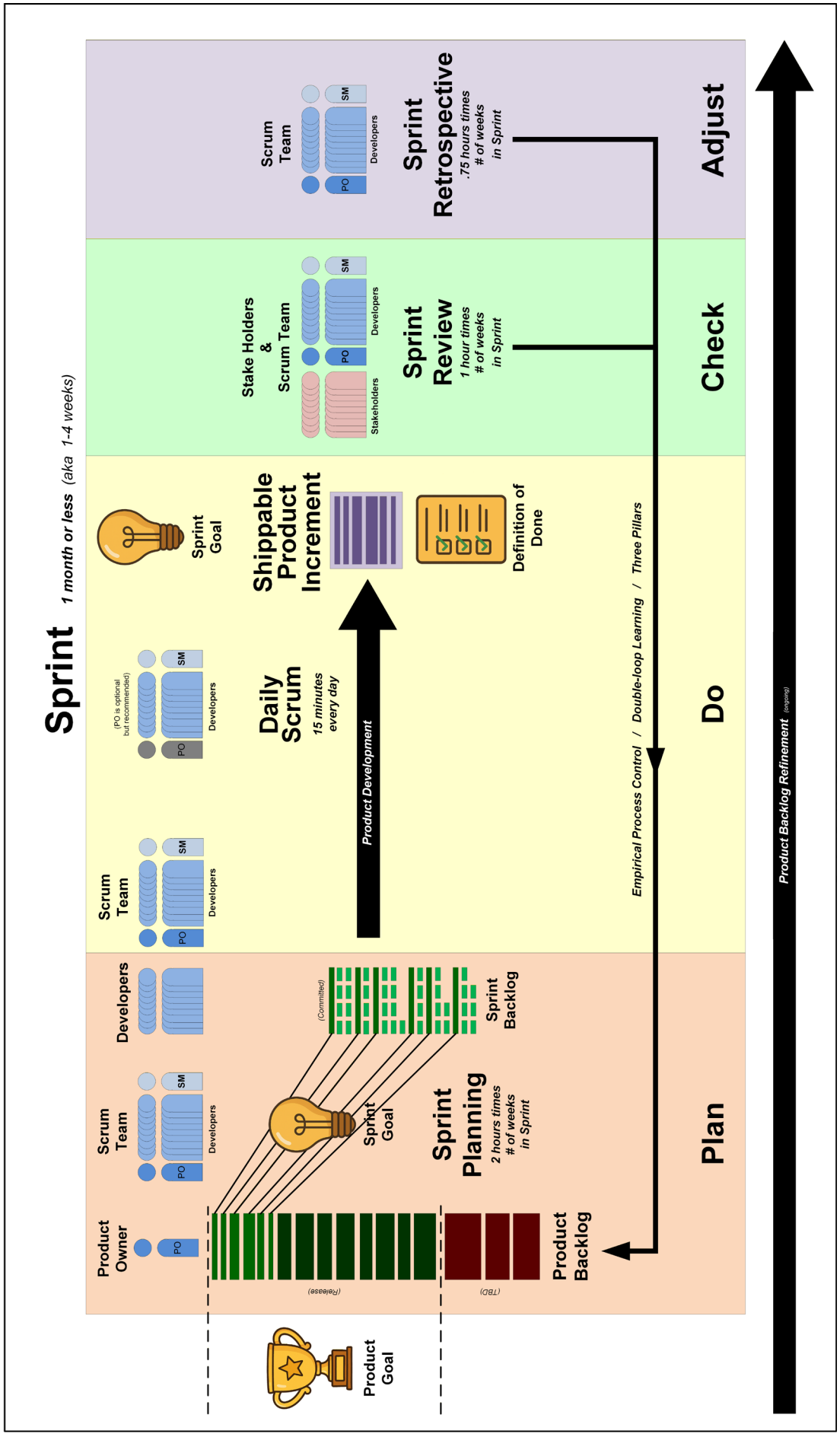
The emphasis is on a few baseline rules that outline a system for ensuring regular feedback loops to raise transparency.

Scrum is a Product Development framework and thus, says very little about Project Management specifically.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum Framework

From The Scrum Guide by Ken Schwaber and Jeff Sutherland

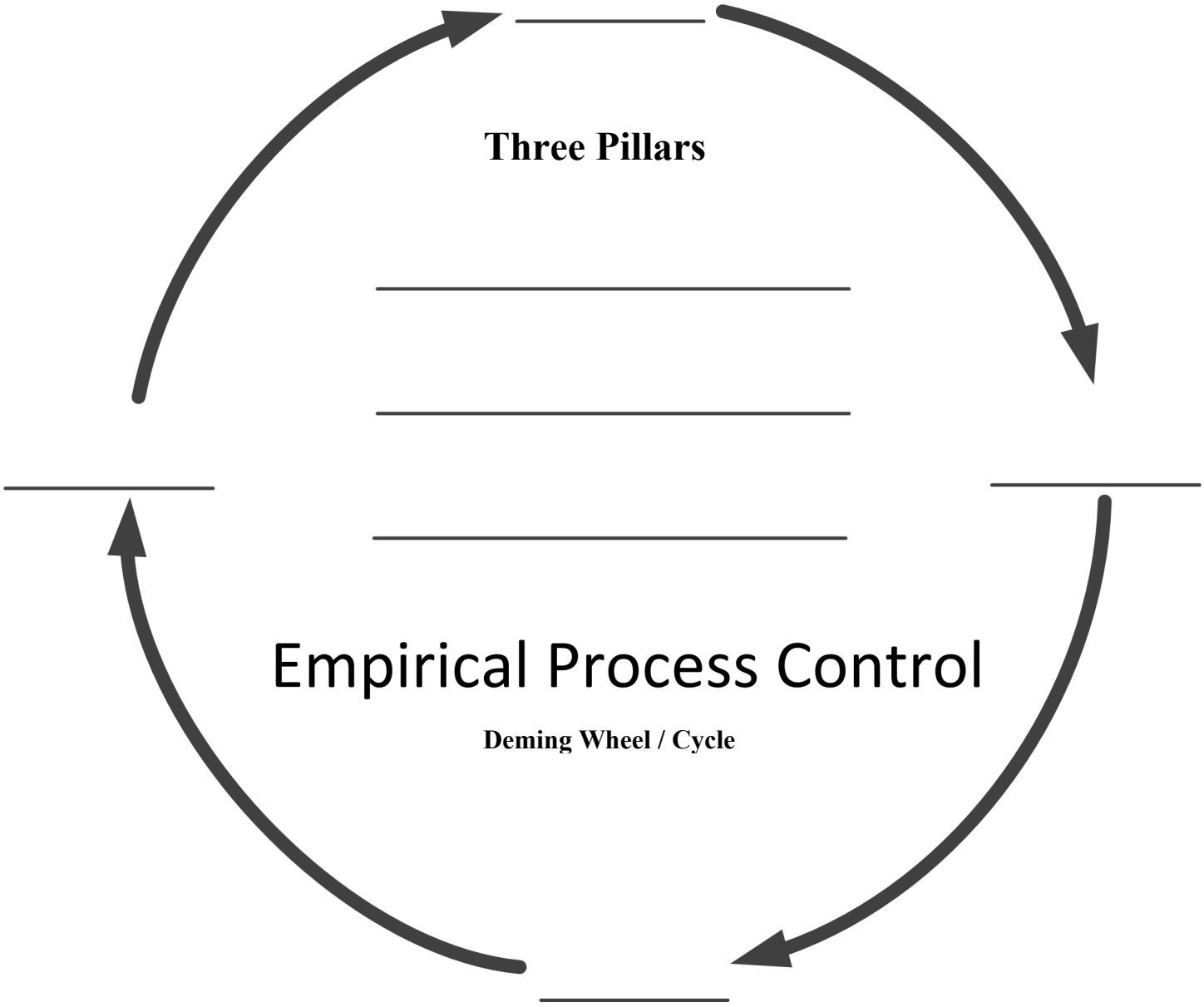


 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Define empiricism and list the three empirical Scrum pillars

Scrum enables incremental and iterative development of products through various cycles or feedback loops which follow the pattern:



empirical - based on, concerned with, or verifiable by observation or experience rather than theory or pure logic

defined - nature, scope, or meaning stated or described in exact terms

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

List the Five Scrum Values

Five Scrum Values

When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone. The Scrum Team members learn and explore those values as they work with the Scrum events, roles and artifacts.

Successful use of Scrum depends on people becoming more proficient in living these five values.

People personally COMMIT to achieving
the goals of the Scrum Team.

The Scrum Team members have COURAGE to do the right thing and work on tough problems.

Everyone FOCUSES on the work of the Sprint and the goals of the Scrum Team.

The Scrum Team and its stakeholders agree to be OPEN about all the work and the challenges with performing the work.

Scrum Team members RESPECT each other
to be capable, independent people.

Working Agreements / Team Norms

Team operating model

Who are we? What do we believe? How do we work?
What does it look like? Where are things documented?
What is our definition of done? What tools do we use?
Who are the backups? Etc.

Sources:
Ken Schwaber. Agile Software Development with Scrum (Upper Saddle River: Prentice Hall, 2002). 148-154.

The Scrum Guide (2020)

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Significant Organizational Change



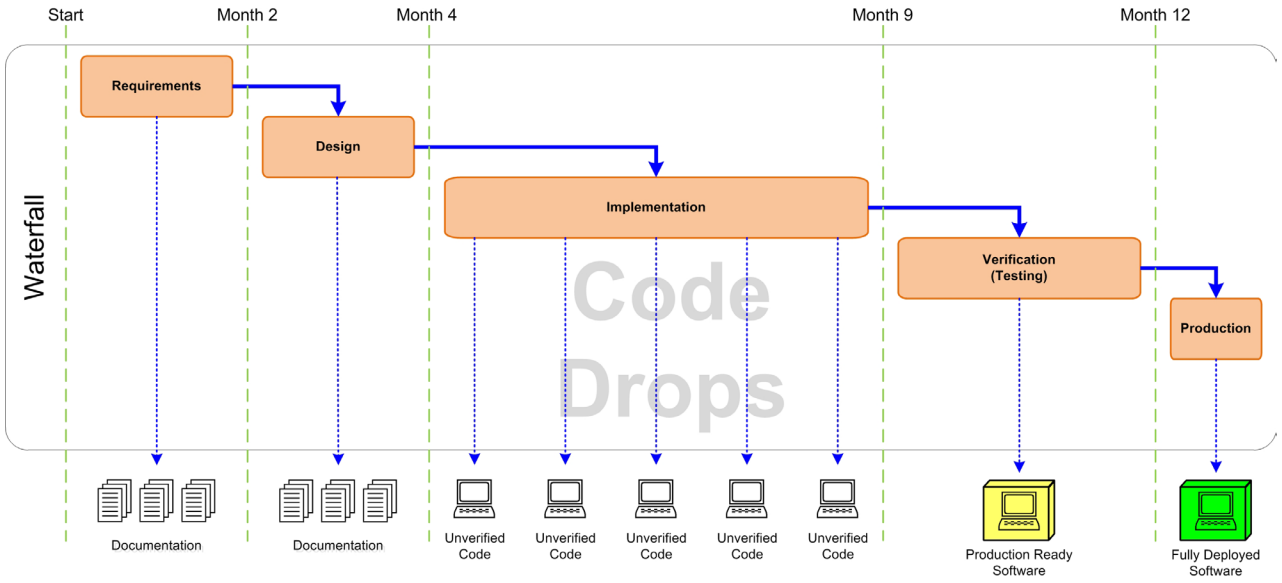
Effective Scrum (Agile) is not simply sprinkled on
top of the organization's current practices.

Current practices MUST change;
which requires a change in mindset, priorities, and behavior.

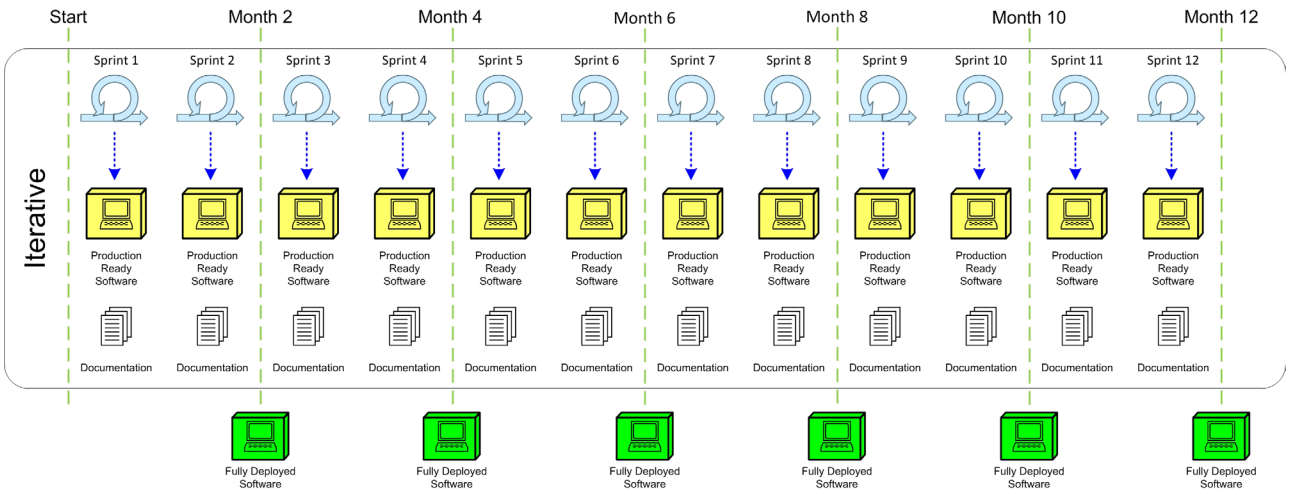
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

List the benefits of an iterative and incremental approach

Example: 12 Month Waterfall Project



Example: 12 Month Scrum Project



Scrum reduces the risk of uncertainty by proving product assumptions

Scrum accelerates the value delivery curve by giving value to the customer early and often

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum Roles

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

ACTIVITY

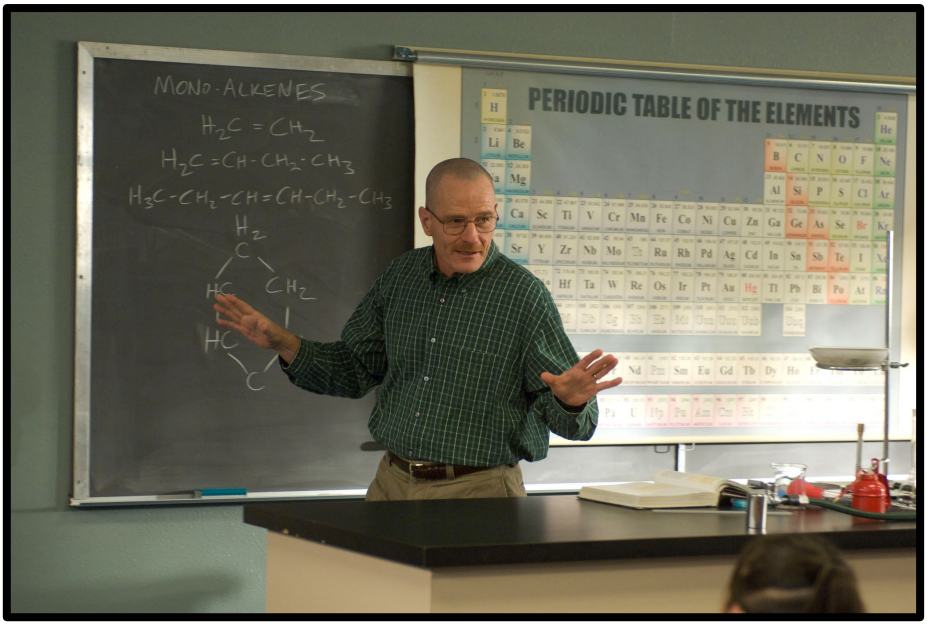
For this exercise, we will break into FOUR teams. Each team will have their own breakout room and Mural space as follows:

- Room 1 – The Scrum Team
- Room 2 – The Product Owner
- Room 3 – The Developers
- Room 4 – The Scrum Master

The goal for your team is to create a presentation for the role that you have been assigned. The presentation should teach the role according to the Learning Objective (LOs) listed in the Mural space; i.e. all the things you say about the role MUST match what the Scrum Guide says AND you need to ensure that you address the LO(s) specifically.

Try to make it fun and interesting with visually stimulating elements, creative delivery, etc. The intent is to give you a feel for what it will be like when you are a CSM teaching this to your teams, your organization, and executives.

You will have **30 minutes** for this exercise. When we come back together as a group, your team should be prepared to present and teach the other classmates the role that you have been assigned.

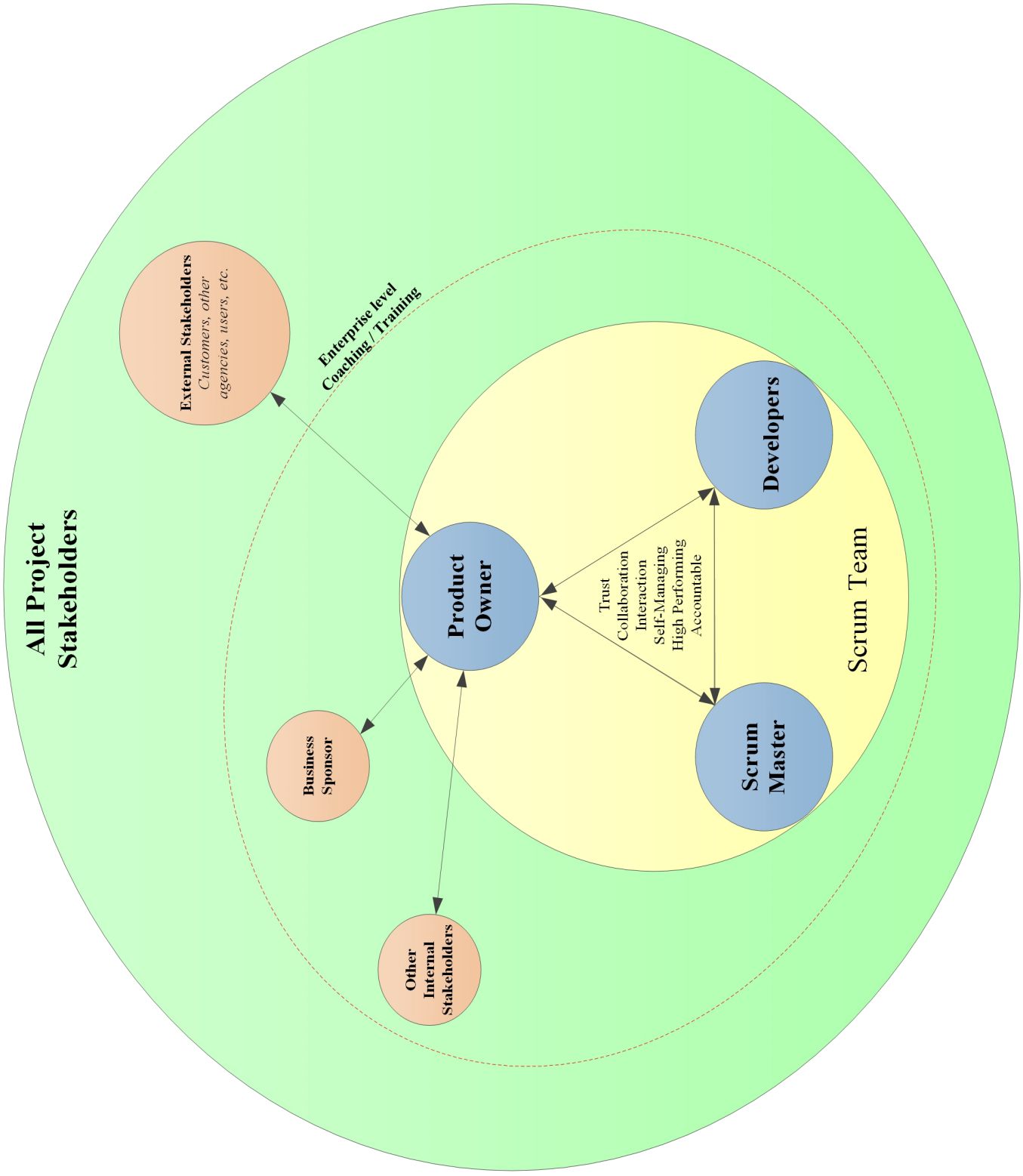


 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

The Scrum Team

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-1.1 Describe the responsibilities and accountabilities of the Scrum Team



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Describe at least three DISADVANTAGES
of shared roles in Scrum

- **Developers serving as ScrumMaster**
- **Developers serving as Product Owner**
- **Product Owner serving as ScrumMaster**

In the REAL WORLD, you have 100% of your attention to give. If you are assigned to two roles, you will never be able to give BOTH your full attention. In the end, BOTH will suffer and remain undeveloped in terms of potential.

"No one can serve two masters, for either he will hate the one and love the other, or he will be devoted to the one and despise the other. You cannot be the ScrumMaster AND Product Owner or ScrumMaster AND a Developer."

Mathew 6:24 ESV*(sort of)*

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Consider the Scrum Team as a Social System

US Government

(how it was designed to work, that is)

Three branches
Checks and balances
Equal

With 3 entities in the social system, there can be a majority opinion versus having stalemate with only 2 entities in the system.

The ScrumMaster and Product Owner should not be the same person because that would be a conflict of interest.

The ScrumMaster should not ALSO be a Developer because that would be a conflict of interest.

Only when each role is fulfilled individually can there be a balanced social system with the capacity for conflict resolution.

The Scrum Guide does not prescribe who does the hiring or assigning of people to teams. Thus, it is beyond the scope of Scrum.

Team Size:
10 or fewer people

The **Ringelmann effect** is the tendency for individual members of a group to become increasingly less productive as the size of their group increases.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Product Owner

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

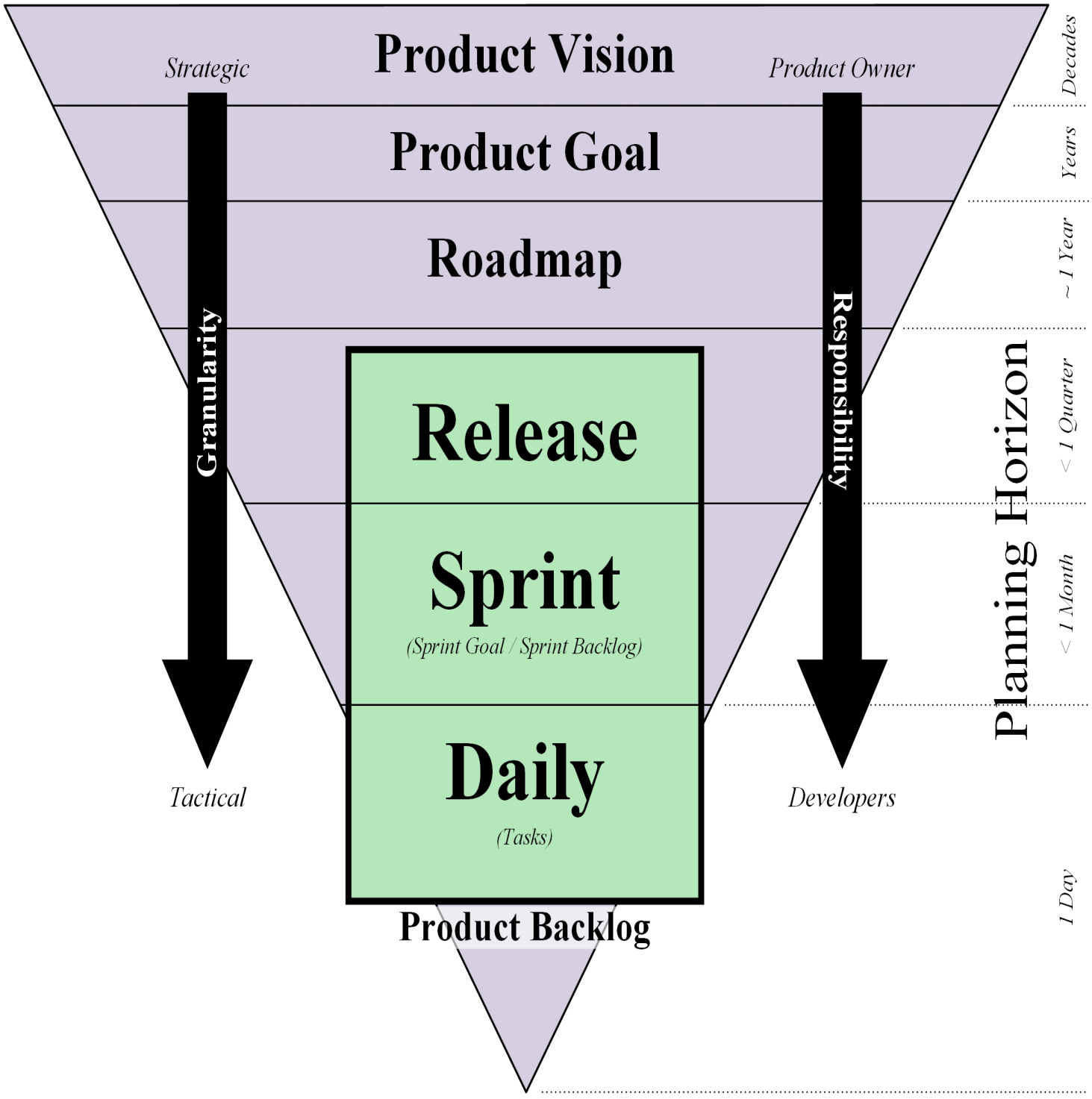
CSM-1.4 Describe the responsibilities and accountabilities of the Product Owner

- Independent authority
- Defining scope for the Developers
- Deciding when to release
- Ongoing visioning
- Ordering of the Product Backlog
- Budgeting
- Determining Release Dates
- Maximizing the value of the Developers' work
- Maximizing the value of the Scrum Team's product



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Six-level Model of Agile Planning

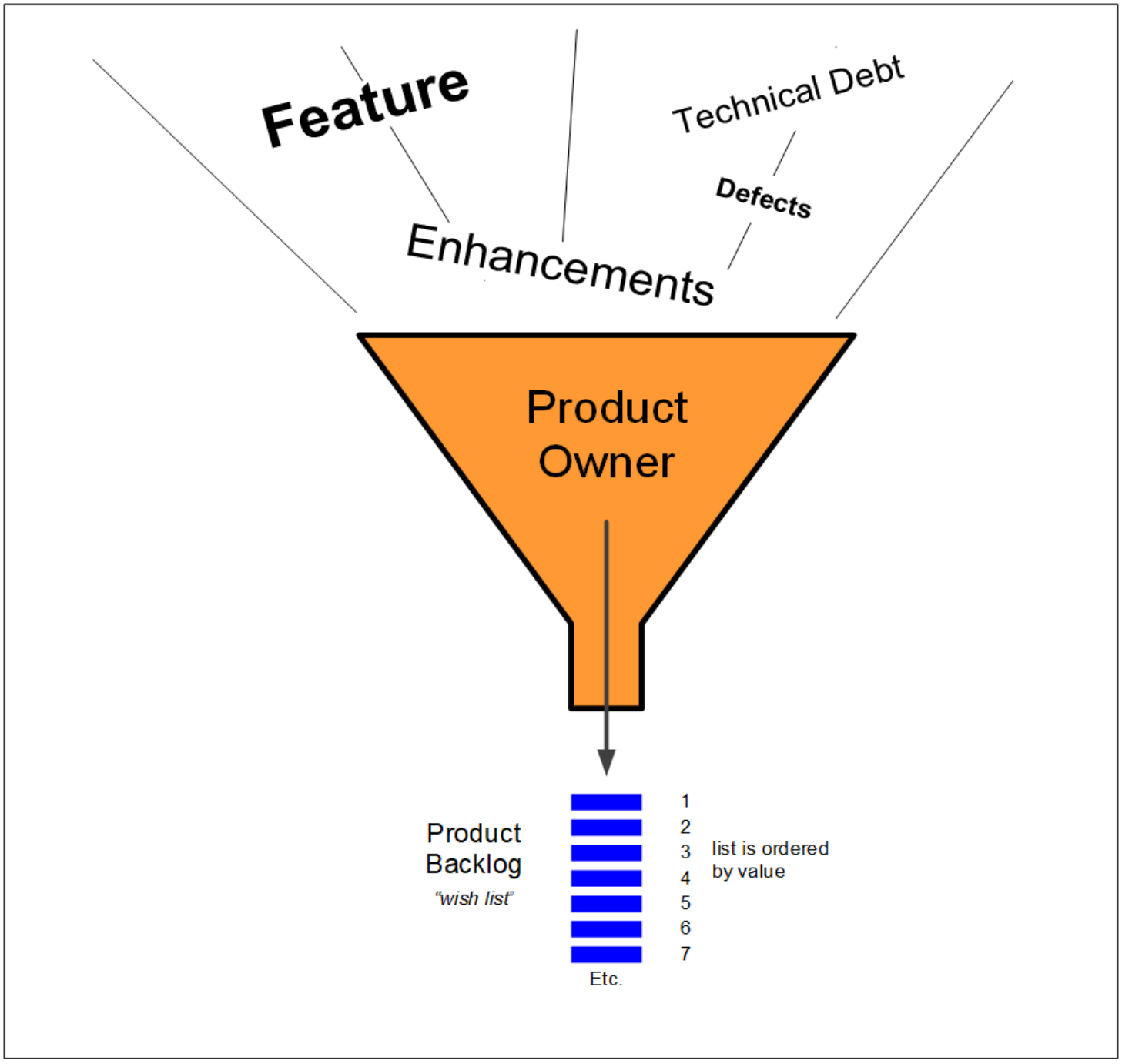


 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

CSM-1.5 Discuss at least two reasons why the Product Owner is a single person and neither a group nor a committee

Product Owner Funnel



The Product Owner funnels ALL work requests intended for a Development Team into the ordered Product Backlog by making trade-offs between the various items.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-1.6 Discuss how and why the Product Owner maintains authority over the Product Backlog while working collaboratively with Developers and stakeholders

Ultimately Accountable
Ultimate Authority

Both are required.
Cannot have one without the others.

The Product Owner...

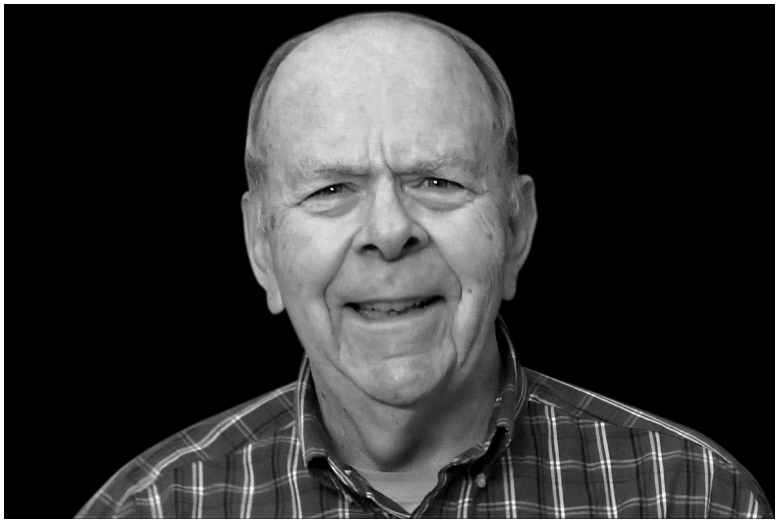
Meets with customer regularly
Tracks finances
Manages contracts
Deals with vendors
Ensures product follows vision
Balances customer (stakeholder) needs with organization needs
Backlog refinement
An amalgam of Product Manager, Project Manager, Business Analyst, Marketing Analyst, Customer Experience Designer, Sales, etc.

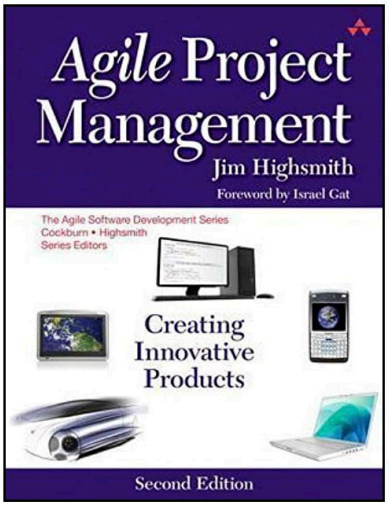
However,
A PRODUCT OWNER CANNOT ALSO BE THE SCRUM MASTER!!

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-3.9 Discuss why Scrum does not have a project manager

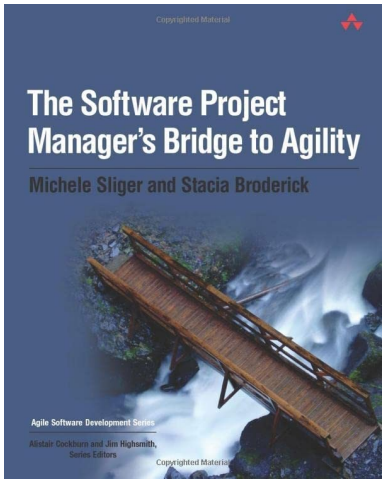
Consider the responsibilities of the
Traditional PM vs. Product Owner

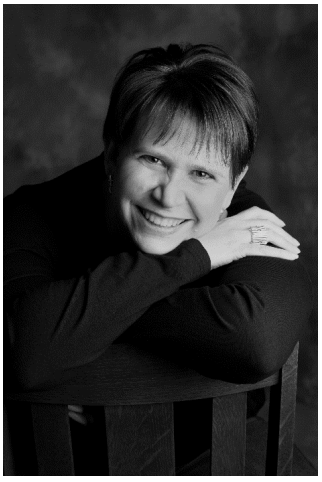




"Process is not a substitute for skill."

-Jim Highsmith







The Software Project Manager's Bridge to Agility
Michele Sliger and Stacia Broderick

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Developers

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

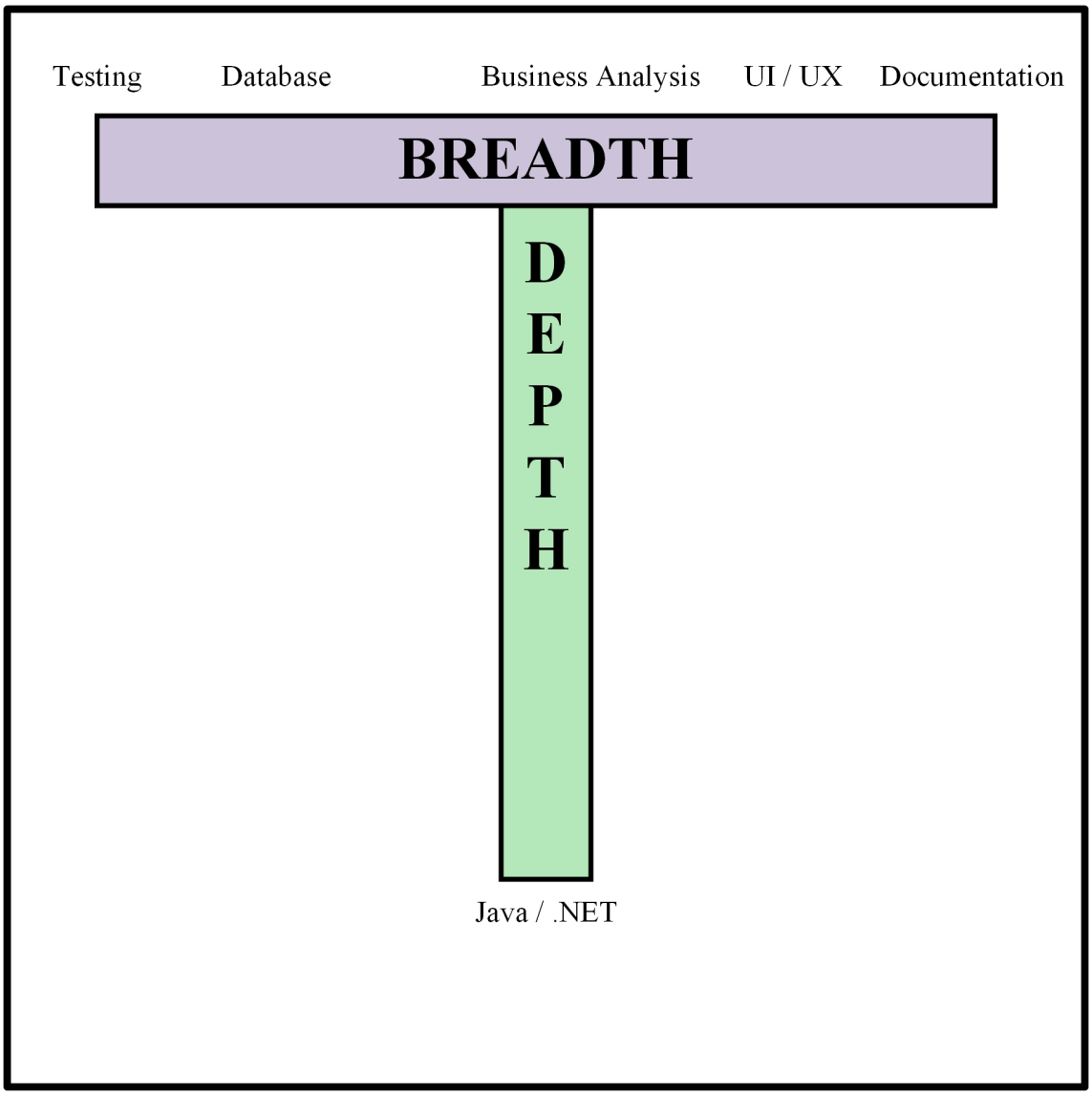
CSM-1.3 Describe the responsibilities and accountabilities of the Developers

- **Understand the "what"**
- **Determine the "how"**
- **Assess complexity**
- **Determine skills necessary to deliver**
- **Delivery of shippable features**
- **Mitigating dependencies**
- **Proposing features**
- **Explaining technical concerns**
- **Providing estimates**
- **Taking initiative**
- **Collaborating with PO**
- **Defining "Done"**

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

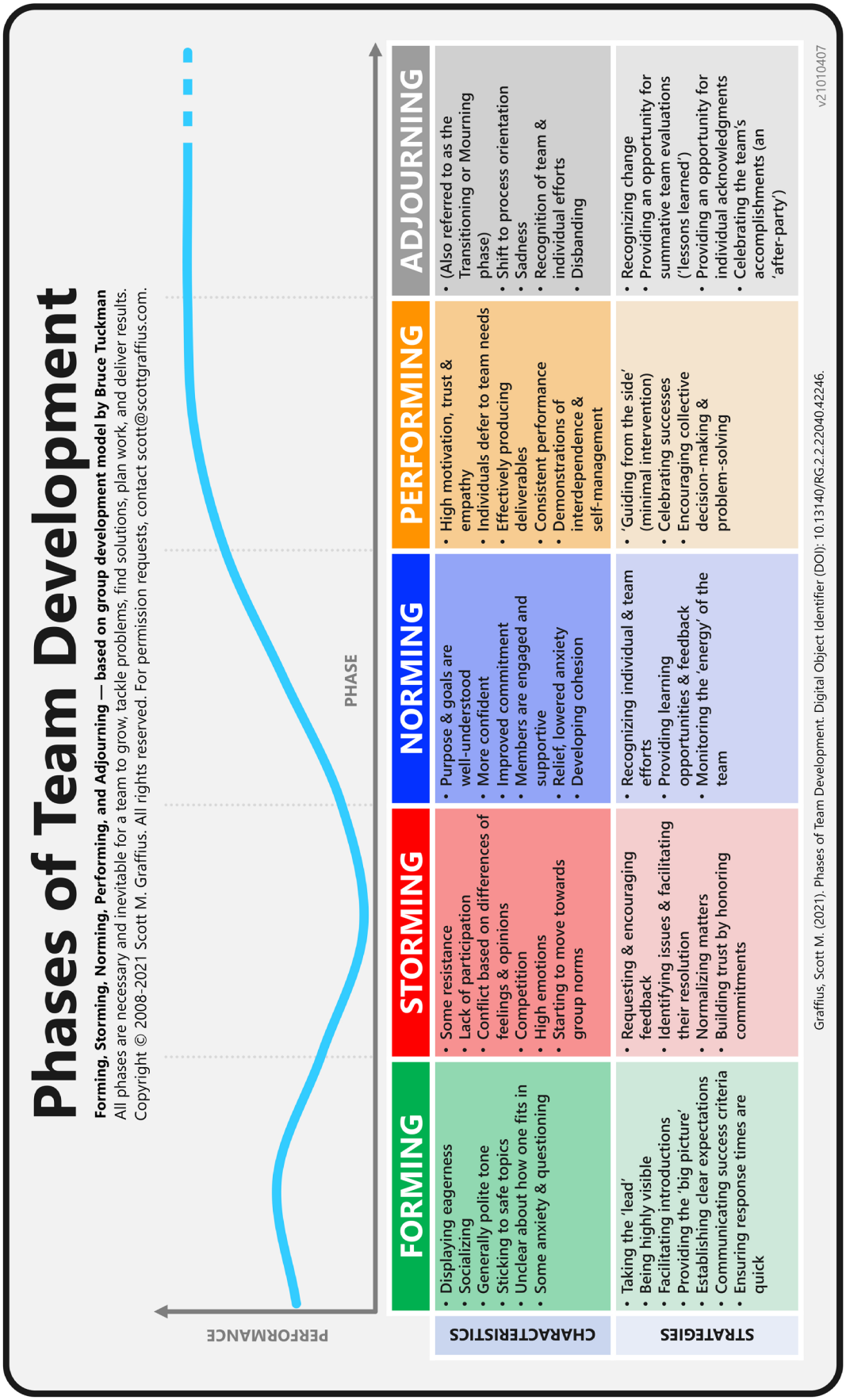
Cross-functional Team

All the skills are there to build (develop) the solution;
i.e. deliver customer value.



T-Shaped Skills

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Scrum Master

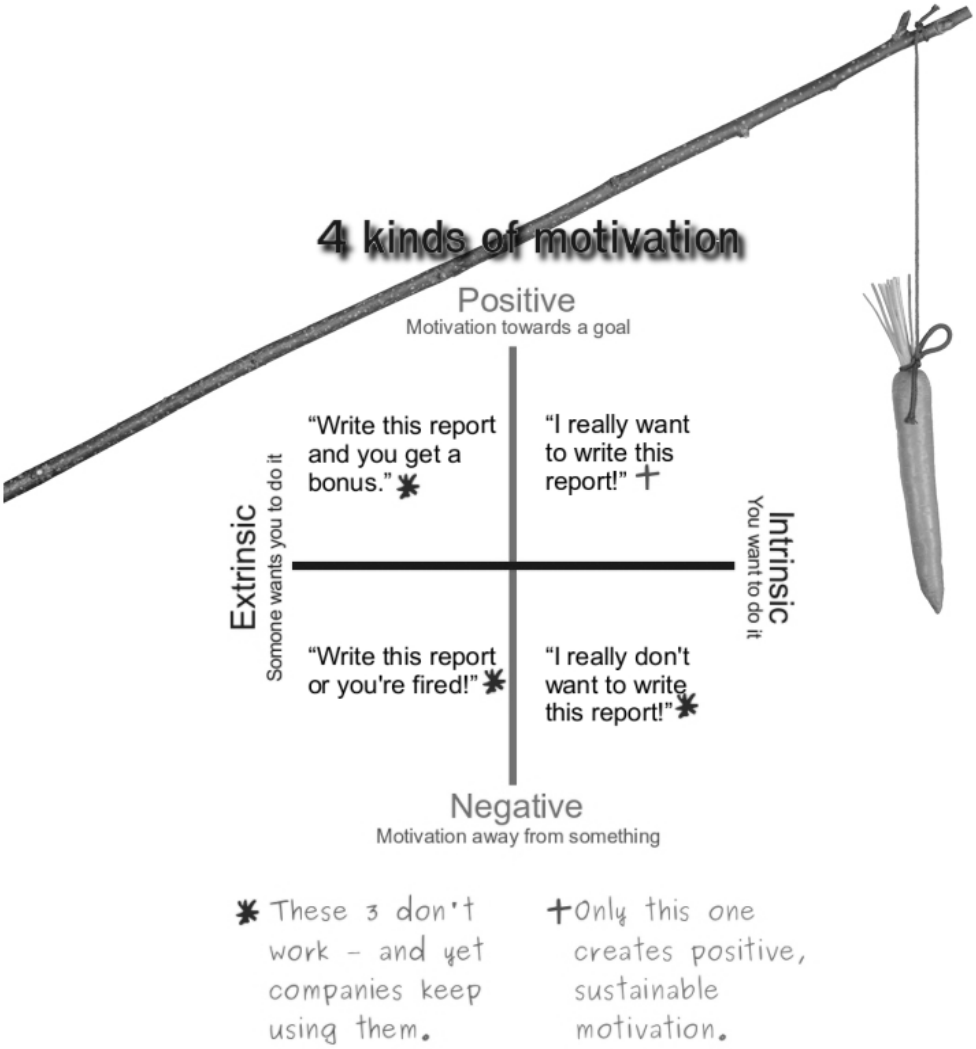
Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-1.2 Describe the responsibilities and accountabilities of the Scrum Master

- **Coach the Scrum Team**
- **Remove impediments**
- **Recognize dysfunction**
- **Ensure that Scrum practices are followed**
- **Provide guidance / advice**
- **Suggest improvements**
- **Conflict resolution**
- **Counseling**
- **Translator (business-tech, tech-business)**
- **Motivator**
- **Servant**
- **Leader, not manager**

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Motivation / Drive



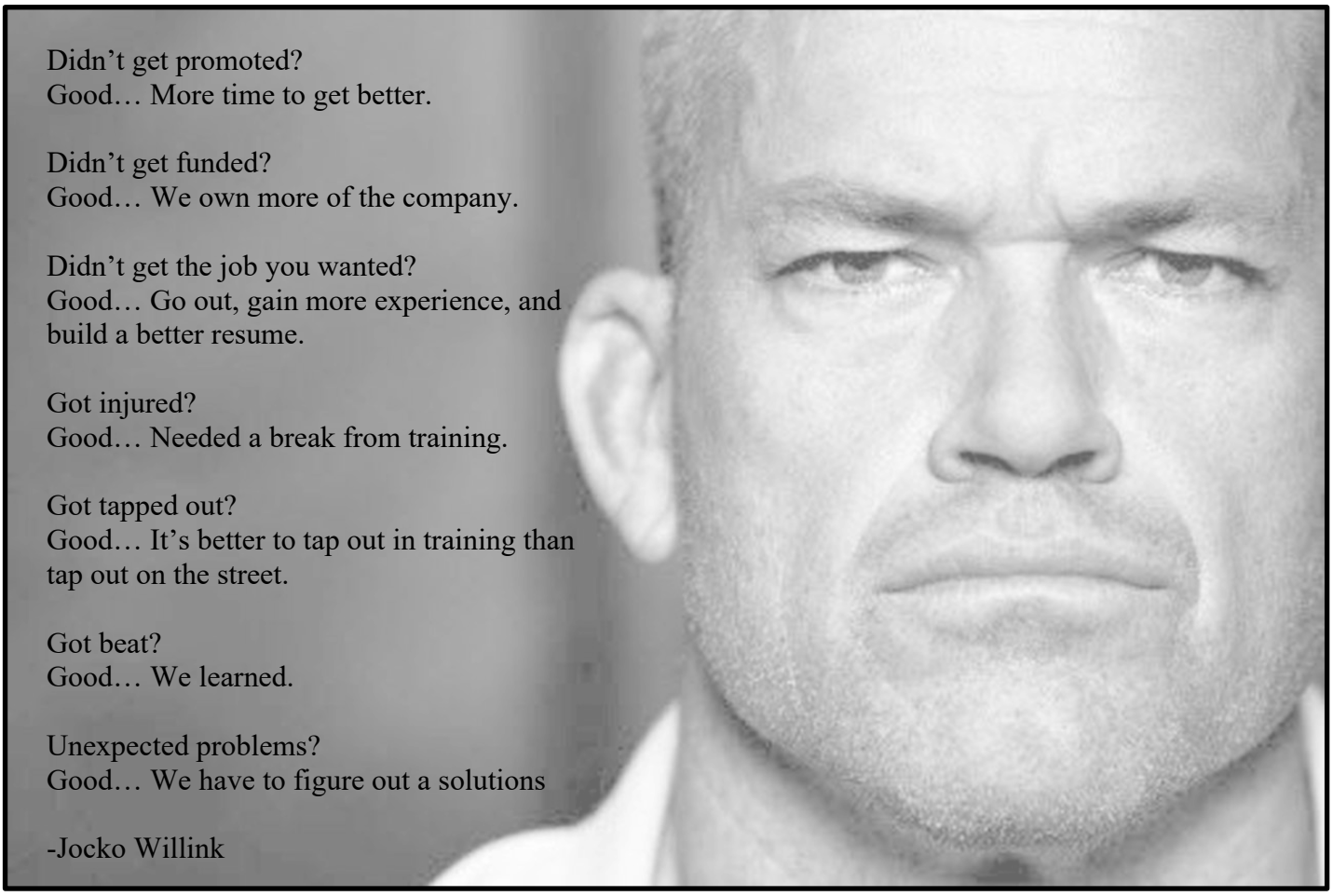
Source: "Why "Motivation by Pizza" Doesn't Work", Alexander Kjerulf, 2006

"The 3 components of motivation are – **autonomy**, **mastery and purpose** ."
-Dan Pink, Drive: The Surprising Truth About What Motivates Us

"The ultimate freedom for creative groups is the freedom to experiment with new ideas. Some skeptics insist that innovation is expensive. In the long run, innovation is cheap. Mediocrity is expensive—and autonomy can be the antidote."
-Tom Kelley, General Manager, IDEO

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

“GOOD.”



Jocko Willink

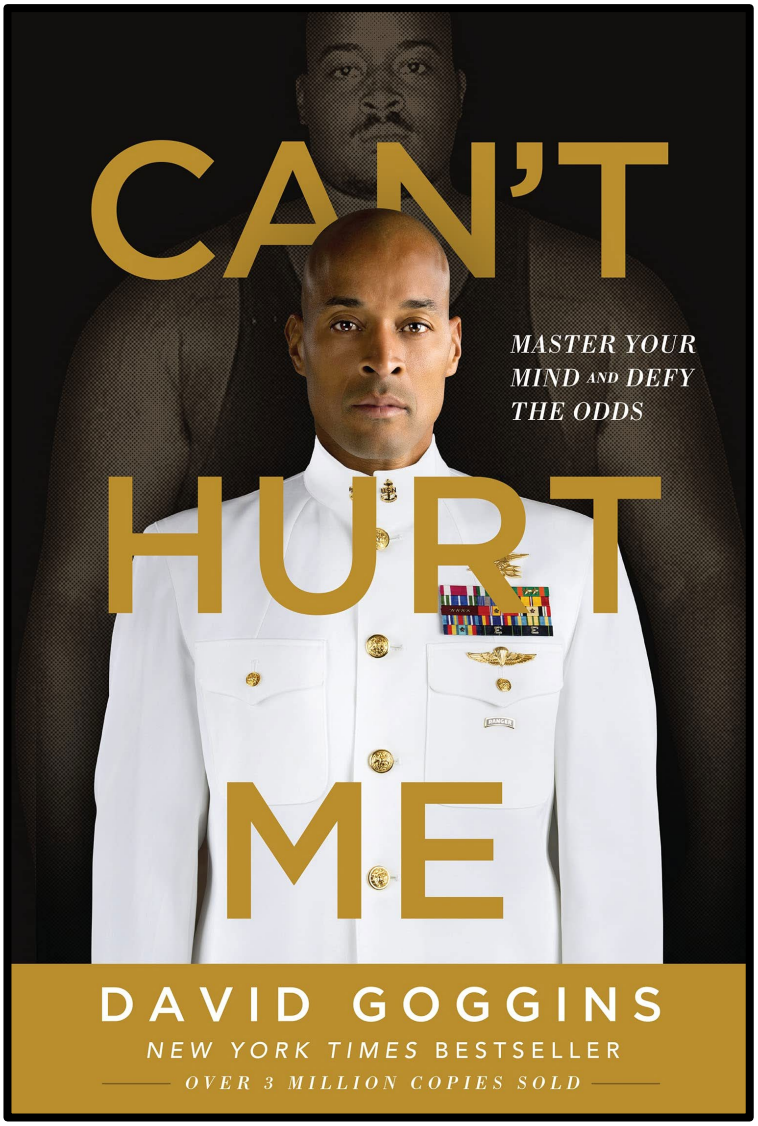
Extreme Ownership

(motivation, integrity, loyalty, characteristics of a servant leader, resourcefulness, FSO-GSD)

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

More on Motivation

"Get rid of your comfort zone."
-David Goggins



(motivation, stop making excuses, NO WHINING, internal locus of control)

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

An Example Checklist for Scrum Masters

Michael James
(mj@seattlescrum.com)
Revised 8 Feb 2022
https://scrummasterchecklist.org

A Full Time Facilitator?

An adequate Scrum Master can handle two or three teams at a time. If you're content to limit your role to organizing meetings, enforcing timeboxes, and responding to the impediments people explicitly report, you can get by with part time attention to this role. Probably nothing catastrophic will happen.

But if you envision a team that succeeds at things they didn't previously realize they could do, consider being a *great* Scrum Master.

A great Scrum Master can handle *one* team at a time.

We recommend one dedicated Scrum Master per team of about six when starting out.

If you haven't discovered all the work there is to do, tune in to your Product Owner, your team, your team's engineering practices, and the organization outside your team. While there's no single prescription for everyone, I've outlined typical things I've seen Scrum Masters overlook. Please mark each box with √, Δ, ?, or N/A, as described on the last page.

Part I -- How Is My Product Owner Doing?

Scrum Masters improve Product Owner effectiveness in many ways.

- ☐ Regardless of the number of teams, is there one (and only one[1]) Product Owner for the product?
- ☐ Is the Product Backlog prioritized according to his/her latest thinking?
- ☐ Are newly discovered requirements and desirements being captured in the Product Backlog? Remember: the backlog is *emergent*.
- ☐ Is the Product Backlog a manageable size? To maintain a manageable number of items, keep things more granular towards the top, with general epics at the bottom. It's counterproductive to overanalyze too far past the top of the Product Backlog. Plans will change in an ongoing conversation between the developing product and the users/customers.
- ☐ Could any requirements (especially those near the top of the Product Backlog) be better expressed as independent, negotiable, valuable, estimable, small, and testable user stories[2]?
- ☐ Have you educated your Product Owner about *technical debt* and how to avoid it? One piece of the puzzle may be to write automated test and refactoring into the definition of "done" for each backlog item.
- ☐ Is the backlog an *information radiator*, immediately visible to all concerned?
- ☐ If you're using an automated tool for backlog management, does everyone know how to use it easily? Sadly, automated management tools usually become anti-collaborative *information refrigerators*.

---

[1] https://seattlescrum.com/Why-Scrum-Isnt-Making-Your-Company-Very-Agile/

[2] https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/

Copyright © 2004-2022 Michael James (MJ). Distribute freely without alteration.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

- [ ] Can you help radiate information by teaching how to create big visible charts?
- [ ] Have you helped your Product Owner organize backlog items into appropriate releases or priority groups?
- [ ] Does everyone know whether the release plan still matches reality? You might try showing everyone Product/Release Burndown Charts[3] after the items have been acknowledged as "done" during every Sprint Review Meeting. Charts showing both the rate of PBIs actually completed and new ones added allow early discovery of scope/schedule drift.
- [ ] Did your Product Owner adjust any release forecasts after the last Sprint Review Meeting? If you want to ship a properly tested product by a fixed date, you must constantly reconsider scope. This probably requires deferring some work for future releases as more important work is discovered. Some products can be released every Sprint, or multiple times per Sprint.

Part II -- How Is My Team Doing?

While the Scrum Master are encouraged to lead by the example of collaborating with team members on their work, there is a risk of getting lost in technical tasks. Consider your primary responsibilities to the team:

- [ ] Is your team in the state of *flow*? Some characteristics of this state[4]:
  - Clear goals (expectations and rules are discernible and goals are attainable, aligning appropriately with one's skill set and abilities).
  - Concentration and focus, a high degree of concentration on a limited field of attention.
  - A loss of the feeling of self-consciousness, the merging of action and awareness.
  - Direct and immediate feedback (successes and failures in the course of the activity are apparent, so that behavior can be adjusted as needed).
  - Balance between ability level and challenge (the activity is neither too easy nor too difficult).
  - A sense of personal control over the situation or activity.
  - The activity is intrinsically rewarding, so there is an effortlessness of action.
- [ ] Do team members seem to like each other, goof off together, and celebrate each other's success?
- [ ] Do team members hold each other accountable to high standards, and challenge each other to grow?
- [ ] Are there issues and opportunities the team isn't discussing because they're too uncomfortable?[5]
- [ ] Have you tried a variety of formats and locations for Sprint Retrospectives?[6]
- [ ] Has the team kept focus on Sprint goals? Consider a mid-Sprint checkup to reassess the Sprint Plan.
- [ ] Does the Sprint taskboard reflect what the team is actually doing? Beware the "dark matter" of undisclosed tasks and tasks bigger than one day's work. Work not related to the Sprint is an impediment to the Sprint.

---

[3] Mike Cohn, *Agile Estimation and Planning*. (2005).

[4] Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience* (1990).

[5] Marshall Rosenberg, *Nonviolent Communication: A Language of Life: Life-Changing Tools for Healthy Relationships* (2003). Also consider enlisting a professional facilitator who can make uncomfortable conversations more comfortable.

[6] Derby/Larson *Agile Retrospectives: Making Good Teams Great* (2006).

Copyright © 2004-2022 Michael James (MJ). Distribute freely without alteration.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

- [ ] Does your team have 3-9 people with a sufficient mix of skills to build a potentially shippable product increment?
- [ ] Is your team's taskboard up to date?
- [ ] Are the team self-management artifacts visible to the team, convenient for the team to use?
- [ ] Are these artifacts adequately protected from meddlers? Excess scrutiny of daily activity by people outside the team may impede team internal transparency and self management.
- [ ] Do team members volunteer for tasks? On my first Scrum team we felt like volunteers who also got paid. If it doesn't feel like that, something's gone wrong somewhere.
- [ ] Has the need for technical debt repayment been made explicit in the definition of *done*, gradually making the code a more pleasant place to work?
- [ ] Are team members leaving their job titles at the door of the team room, collectively responsible for all aspects of agreed work (testing, user documentation, etc.)?

Part III -- How Are Our Engineering Practices Doing?

- [ ] Does your system in development have a "push to test" button allowing anyone (same team or different team) to conveniently detect when they've caused a regression failure (broken previously-working functionality)? Typically this is achieved through the xUnit framework (JUnit, NUnit, etc.).
- [ ] Do you have an appropriate balance of automated end-to-end system tests (a.k.a. "functional tests") and automated unit tests?
- [ ] Is the team writing both system tests and unit tests in the same language as the system they're developing? Collaboration is not enhanced by proprietary scripting languages or capture playback tools that only a subset of the team knows how to maintain.
- [ ] Has your team discovered the useful gray area between system tests and unit tests?
- [ ] Does a continuous integration[7] server automatically sound an alarm when someone causes a regression failure? Can this feedback loop be reduced to hours or minutes? ("Daily builds are for wimps." -- Kent Beck)
- [ ] Do *all* tests roll up into the continuous integration server result?
- [ ] Have team members discovered the joy of continuous design and constant refactoring[8], as an alternative to Big Up Front Design? Refactoring has a strict definition: changing internal structure without changing external behavior. Refactoring should occur continuously, or at least several times per hour. Refactor whenever there is duplicate code, complex conditional logic (visible by excess indenting or long methods), poorly named identifiers, excessive coupling between objects, etc. Refactoring with confidence is only possible with automated test coverage. Neglecting refactoring makes it hard to change the product in the future, especially since it's hard to find good developers willing to work on bad code.
- [ ] Does your definition of "done" for each Product Backlog Item include full automated test coverage and refactoring? Learning Test Driven Development (TDD) increases the probability of achieving this.

---

[7] http://www.martinfowler.com/articles/continuousIntegration.html

[8] Martin Fowler, *Refactoring: Improving the Design of Existing Code* (1999).

Copyright © 2004-2022 Michael James (MJ). Distribute freely without alteration.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

- [ ] Are team members pair programming or mob programming most of the time? This can dramatically increase code maintainability and reduce defect rates. It challenges people's boundaries and sometimes seems to take longer (if we measure by lines of code rather than shippable functionality). Lead by example by initiating paired workdays or mobbing sessions with team members. Some of them will start to prefer working this way.

Part IV -- How Is The Organization Doing?

- [ ] Is the appropriate amount of inter-team communication happening? “Scrum of Scrums” is only one way to achieve this, and rarely the best.[9]
- [ ] Are teams able to produce working features, even spanning architectural boundaries?[10]
- [ ] Are you conducting an *Overall Retrospective*[11] to resolve cross-team, organizational and systemic problems within the organization?
- [ ] When appropriate, are the organizational impediments pasted to the wall of the development director's office? Can the cost be quantified in dollars, lost time to market, lost quality, or lost customer opportunities? (But learn from Ken Schwaber's mistakes: "A dead Scrum Master is a useless Scrum Master."[12])
- [ ] Is your organization one of the few with career paths compatible with the collective goals of your teams? Answer "no" if there's a career incentive[13] to do programming or architecture work at the expense of testing, test automation, or user documentation.
- [ ] Has your organization been recognized by the trade press or other independent sources as one of the best places to work, or a leader in your industry?
- [ ] Are you creating a *learning organization*?

Conclusion

If you can check off most of these items and still have time left during the day, I’d like to hear from you.

There’s no canned formula for creating human ingenuity. This paper lists points which may, or may not, help in your situation.

Once you start to realize what you could do to make a difference, you may find yourself afraid to do it. This is a sign you're on the right track.

---

[9] See https://less.works/less/framework/coordination-and-integration/ for alternatives.

[10] https://less.works/less/structure/feature-teams

[11] https://less.works/less/framework/overall-retrospective/

[12] Ken Schwaber, *Agile Project Management with Scrum* (2004)

[13] Alfie Kohn, *Punished By Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes* (1999)

Copyright © 2004-2022 Michael James (MJ). Distribute freely without alteration.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Organizational Impediment Form

Surface issue:

Root cause (Use five times "Why?"):

Business Impact:

Emotional Impact:

Clear, actionable request:

Organizational Impediment Form

Surface issue:

Root cause (Use five times "Why?"):

Business Impact:

Emotional Impact:

Clear, actionable request:

Copyright © 2004-2022 Michael James (MJ). Distribute freely without alteration.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

INSTRUCTIONS

If you have received this checklist as a training assignment and your current (or most recent) employer has been attempting anything like Scrum, please apply this to what you've seen there. Mark each item with one of the following:

√	(for "doing well")
Δ	(for "could be improved and I know how to start")
?	(for "could be improved, but how?")
N/A	(for "not applicable" or "would provide no benefit")

Or, if your current (or most recent) employer has not been attempting anything like Scrum, mark each item with one of the following:

√	(for "doing well" or "would be easy to do well")
Δ	(for "would be a challenge and I know how to start")
?	(for "would be a challenge and I don't know how to start")
N/A	(for "not applicable" or "would provide no benefit")

When all items are marked, declare 2-6 organizational impediments on the attached Organizational Impediment Forms, whether or not they're derived from this checklist. Choose impediments you have at least 1% hope of changing.

Copyright © 2004-2022 Michael James (MJ). Distribute freely without alteration.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

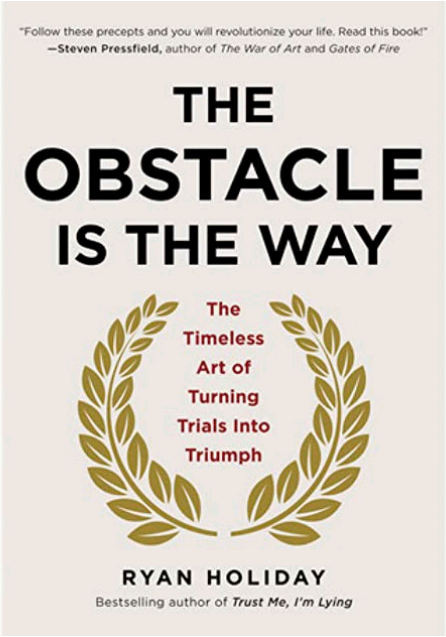
*"Put your head down and **work hard**. Never wait for things to happen, make them happen for yourself through hard graft and not giving up."*

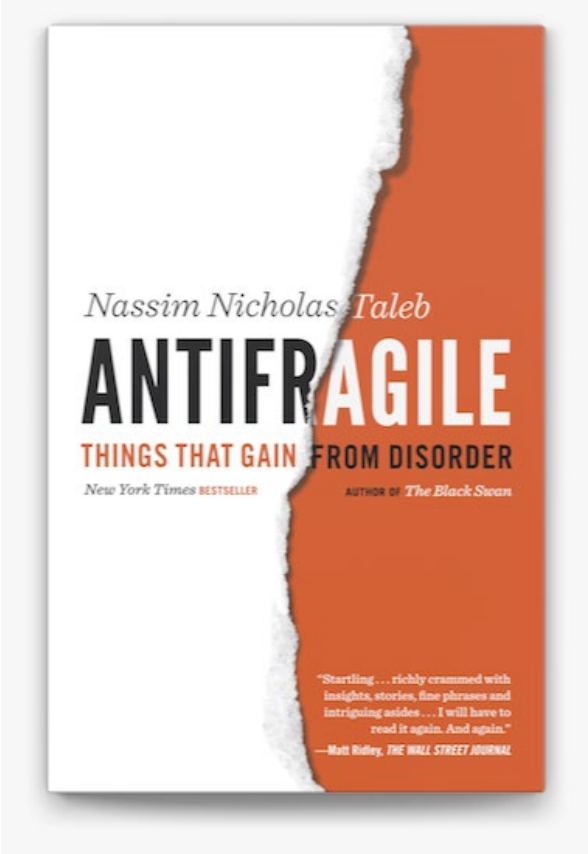
-Gordon Ramsay



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

More great books on
“Stop whining and get your ass in gear!!”





 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum Events and Activities

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

ACTIVITY

For this activity we will be splitting up into 6 teams as follows:

- Room 1 – The Sprint
- Room 2 – Sprint Planning
- Room 3 – Daily Scrum
- Room 4 – Backlog Refinement
- Room 5 – Sprint Review
- Room 6 – Sprint Retrospective

As with the previous exercises, each team will have its own breakout room and Mural space. Each team will be responsible for discussing and putting together a presentation for the event / activity that they have been assigned.

Please make sure that your team is adhering to the Scrum Guide and that the LOs included on your Mural space are covered for that event / activity.

You will have **30 minutes** to work on the presentation. When we come back together, your group should be ready to present.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-1.7 Identify at least one example of how a Scrum Team could inspect and adapt to increase transparency at each of the Scrum events

CSM-1.11 Describe at least three possible effects of skipping the Sprint Retrospective

CSM-1.12 Explain how Developers conduct a Daily Scrum

CSM-1.13 Discuss at least three ways the Daily Scrum differs from a status meeting and why the various constraints exist to support the Developers

CSM-1.14 Explain under what conditions a Sprint could be terminated prematurely

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

The Sprint

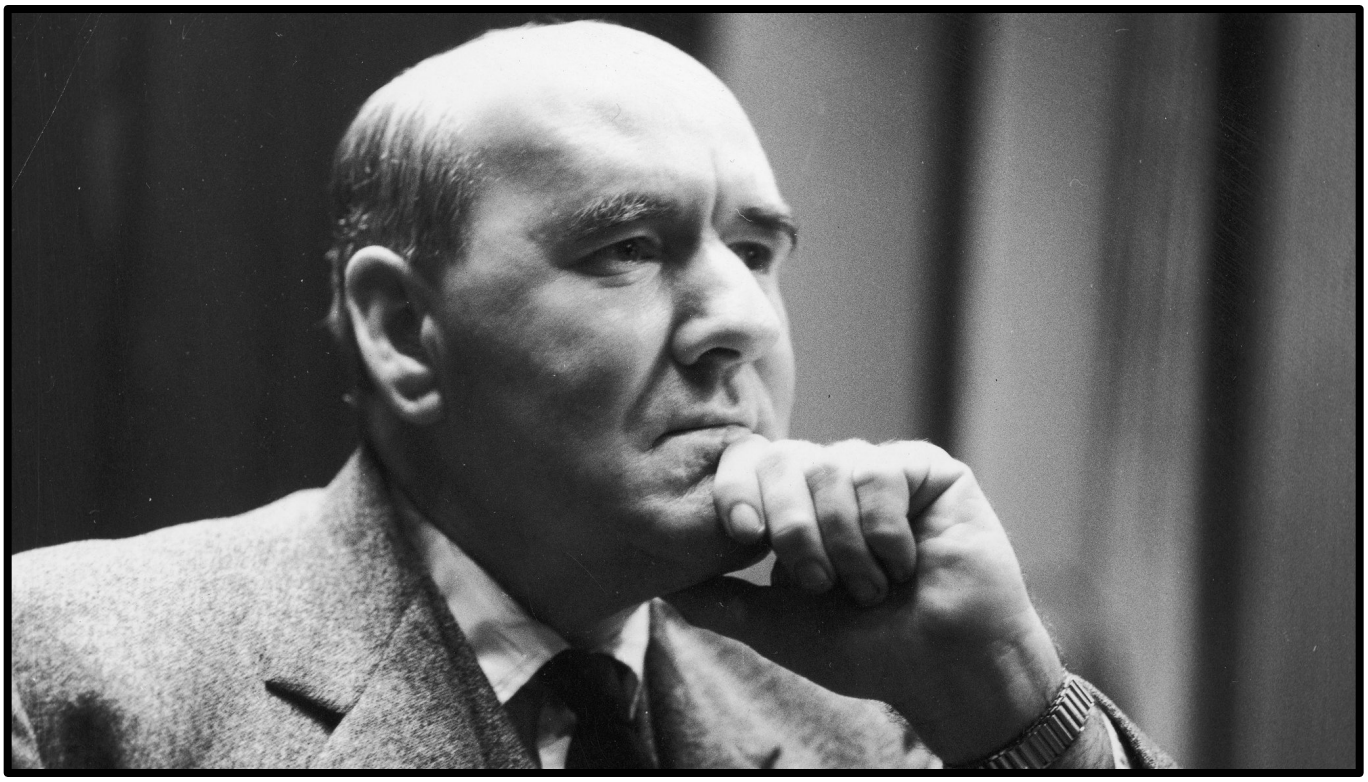
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Benefits of timeboxing

Parkinson's Law

*"**WORK EXPANDS so as to fill the time available for its completion**. General recognition of this fact is shown in the proverbial phrase 'It is the busiest man who has time to spare.' Thus, an elderly lady of leisure can spend the entire day in writing and dispatching a postcard to her niece at Bognor Regis. An hour will be spent in finding the postcard, another in hunting for spectacles, half an hour in a search for the address, an hour and a quarter in composition, and twenty minutes in deciding whether or not to take an umbrella when going to the mailbox in the next street. The total effort that would occupy a busy man for three minutes all told may in this fashion leave another person prostrate after a day of doubt, anxiety, and toil."*

- C. Northcote Parkinson



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Feedback Loops

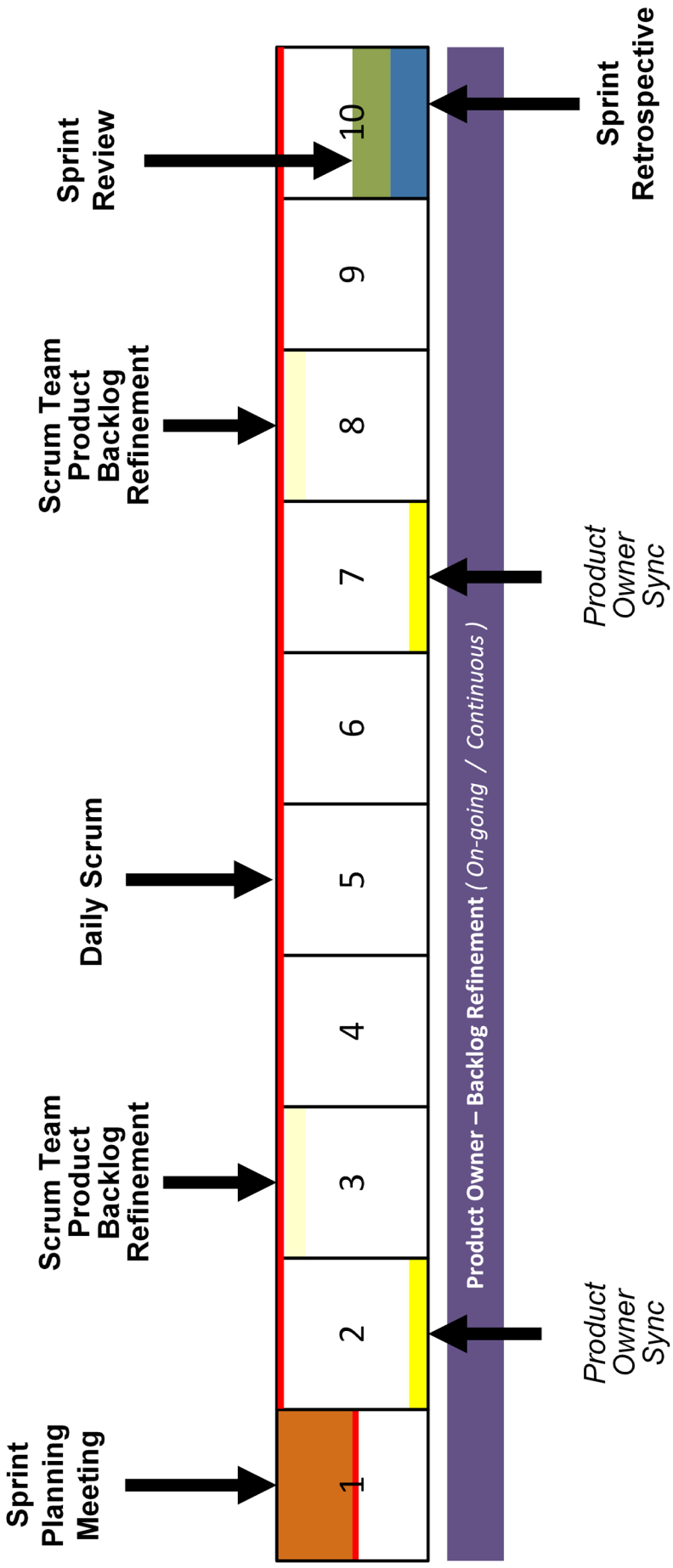


Sailing from San Diego to Hawaii

(just point at 252º and sail for 14 days, right?)

The shorter the Sprint, the more frequently
we are inspecting and adapting.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



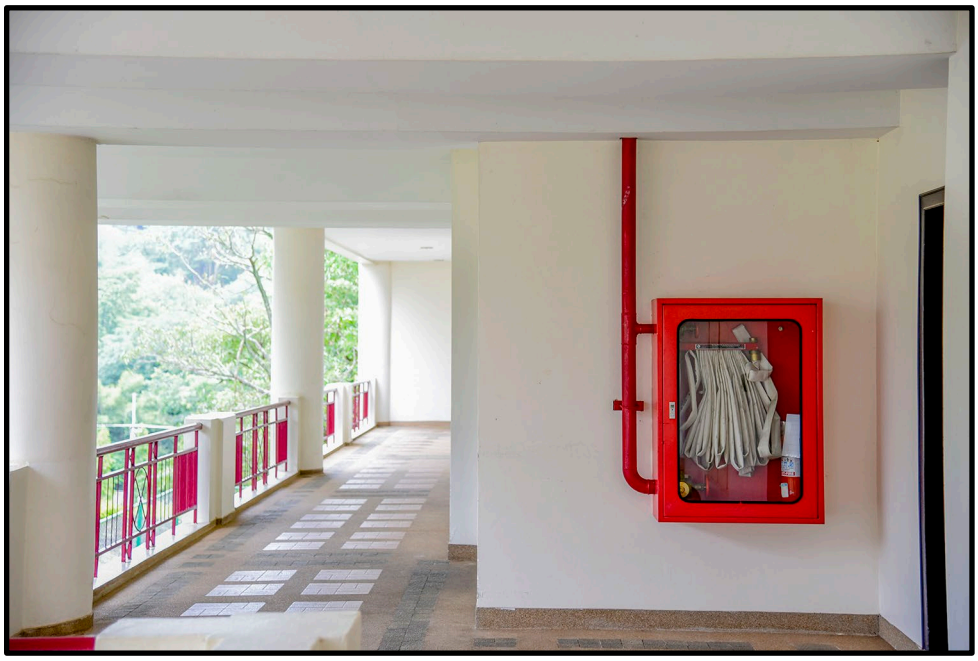
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

CSM-1.14 Explain under what conditions a Sprint could be terminated prematurely

Catastrophic Production Defect

Customer doesn't want anything planned



Canceling the Sprint almost NEVER happens… almost.

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Describe at least two reasons why the scope and duration of a sprint are fixed

Commitment and Discipline



"A deadline has a wonderful way of concentrating the mind."

— Professor Moriarty, *Star Trek: The Next Generation* ("Ship in a Bottle")

Examples:

- Promotes the successful delivery of the Sprint Goal

- Supports the Scrum Team to learn how to deliver valuable increments iteratively

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Define the outcome of every sprint and
describe at least three reasons why that is important

Shippable Product Increment (SPI)

A releasable product increment that adheres to the Definition of Done.

SHIP – ABLE = “Able to be shipped”

Nothing left to do but put it into production

Meets the DoD

Definition of Done

Is valuable and desirable to the customer (i.e. Product Owner)

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Sprint Planning

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Input

Product Backlog Items

(refined)

Sprint Goal

(proposed)

Outputs

Sprint Goal

(confirmed)

Sprint Backlog

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Describe at least three responsibilities for the Developers, Product Owner, and ScrumMaster during Sprint Planning

Planning Meeting

Developers

- Decide how much work is brought into the sprint
- Offer a forecast (COMMIT)
- Create Sprint Backlog items

Product Owner

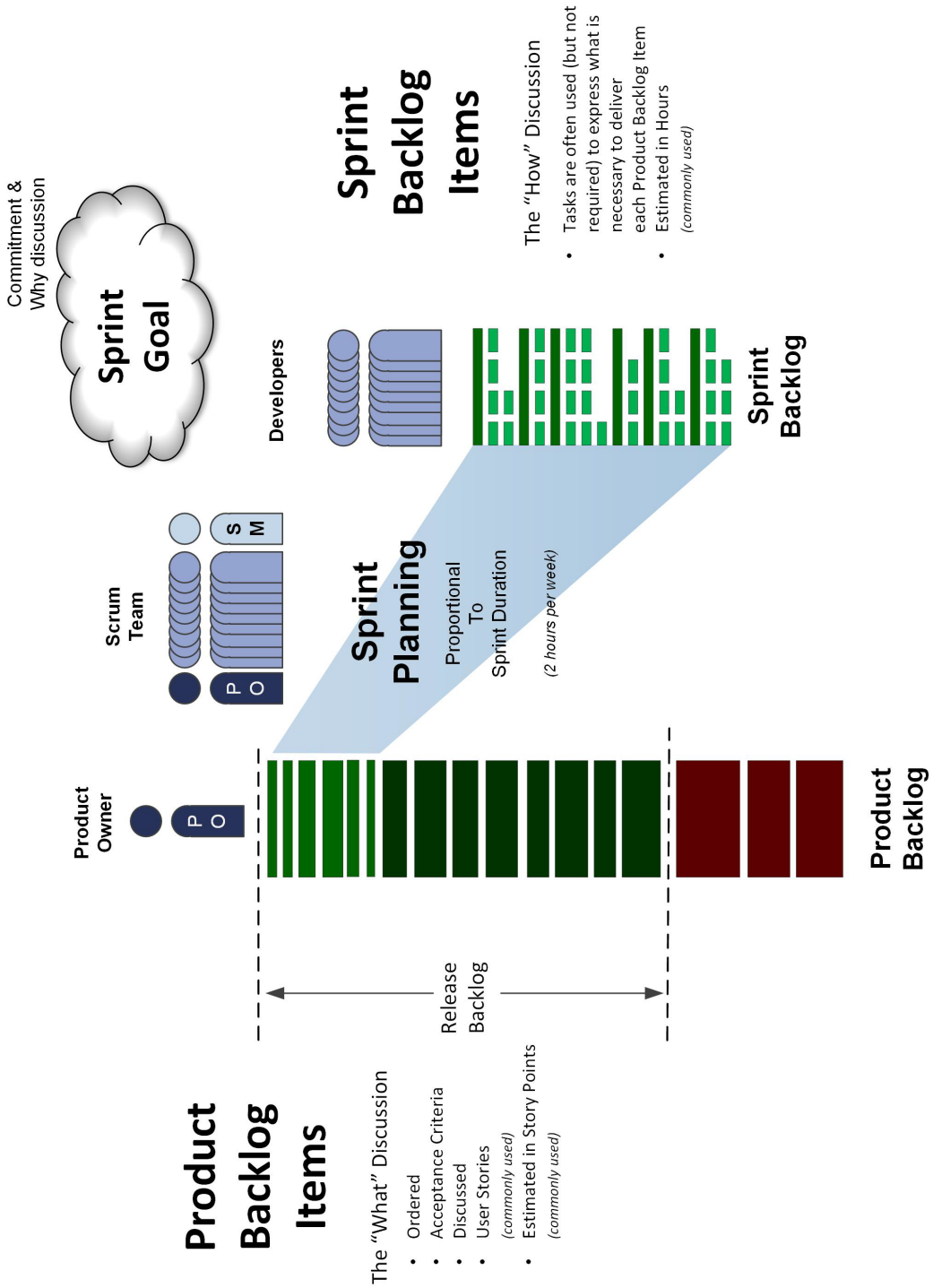
- Offer the Sprint Goal
- Provide clarification
- Negotiate scope

ScrumMaster

- Facilitate the dialogue between the team and the Product Owner
- Maintain the timebox
- Ensure that the Developers neither overcommit nor under-commit

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Discuss the focus of the activities of the Product Owner and Developers during the three topics of Sprint Planning: What? How? Why?



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Give an example of a Sprint Goal and
identify at least two benefits of having a Sprint Goal.

SMART Goals

S – Specific
M – Measurable
A – Achievable
R – Relevant
T – Time-boxed

A Product Goal can roughly be equated to a Release Goal since it represents an increment of value that we are delivering to a customer.

Helps identify alignment of Sprint Backlog items to each other; i.e. not just a collection of random PBIs.

Keeps scrum team focused
throughout the sprint.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Discuss at least three negative impacts that arise when the Scrum Team disregards one or more of the elements of Sprint Planning.



"What are we doing here??!!?"

- Provides greater context for the work
- Helps stakeholders understand why they are being asked to participate in a Sprint Review
- Mitigates the risk that we may be working on things that are NOT valuable

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Daily Scrum

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-1.12 Explain how Developers conduct a Daily Scrum
CSM-1.13 Discuss at least three ways the Daily Scrum differs from a status meeting and why the various constraints exist to support the Developers

- **No more than 15 minutes**
- **Meets every day**
- **Primarily for the Developers to organize themselves for the day**
- **Opportunity to collaborate on the day-to-day, task-level work**
- **Monitor daily progress**
- **Discussion, not a simply "report-out"**



STOP CALLING IT "THE STAND-UP"

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Developers

- Answer the three questions *(optional)*
- Review their progress toward the Sprint Goal
- Update the Sprint Backlog

Product Owner

- Provide clarification
- Offer early feedback
- Give the team the freedom to organize their own work

ScrumMaster

- Teach the Developers how to run the Daily Scrum
- Offer observations, not solutions
- Facilitate the conversation if necessary

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Three Questions

Historically, the three OPTIONAL questions associated with the Daily Scrum agenda were…



"STOP! Who would crosseth the Bridge of Death must answer me these questions three, ere the other side he see..."

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Daniel's Single Question Format:

"Do we have everything we need to go forward and kick ass in the next 24 hours?"



"I just have ONE question, ma'am..."

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Product Backlog Refinement

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items.

This is an ongoing activity to add details,
such as a description, order, and size.

Attributes often vary with the domain of work.

The Developers who will be doing the work
are responsible for the sizing.

The Product Owner may influence the Developers
by helping them understand and select trade-offs.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

STOP CALLING IT "GROOMING"

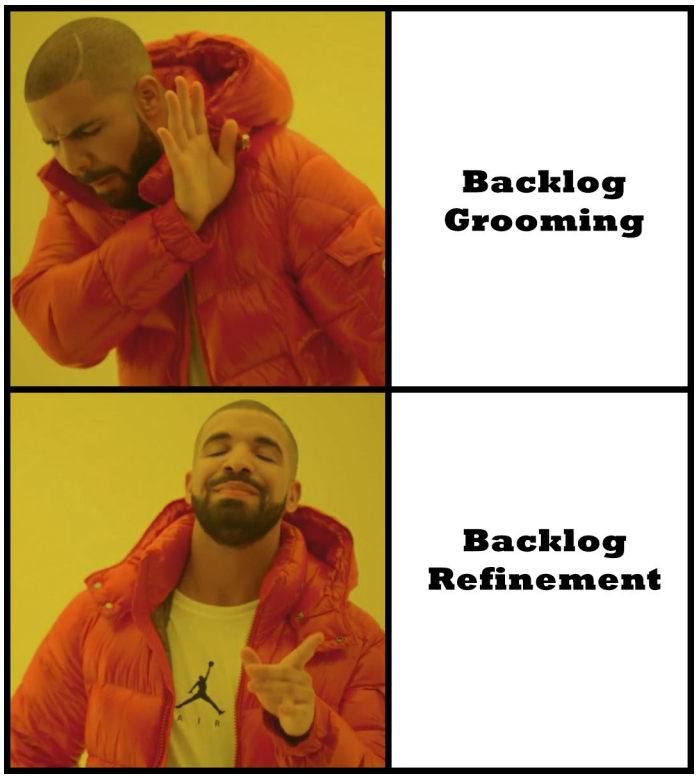
We REFINE the Product Backlog.
We do not GROOM it.

groom

- To clean and brush (an animal)
- To remove dirt and parasites from the skin, fur, or feathers of (another animal)
- To win the confidence of (a victim) in order to a commit sexual assault on him or her

refine

- To become free of impurities
- To acquire polish or elegance
- To use precise distinctions and subtlety in thought or speech
- To make or become free from coarse characteristics; make or become elegant or polished
- To make more fine, subtle, or precise
- To become pure
- To become more elegant or polished
- To make fine distinctions in thought or language



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Sprint Review

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

List the participants of the Sprint Review and describe at least two responsibilities for the Developers, Product Owner, ScrumMaster, and Stakeholders during the Sprint Review.

Developers

- Demonstrate the results of the sprint
- Respond to questions

Product Owner

- Explain which Product Backlog items have been "Done" and which have not
- Track total work remaining toward the release

ScrumMaster

- Ensure that the event takes place
- Maintain the timebox

Stakeholders

- Provide feedback
- Help resolve impediments

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Explain at least four of the review activities that take place during the Sprint Review that pertain to work beyond what has been completed in the sprint, for example: time line, budget, potential use in the marketplace, Product Backlog, release schedule.

- Sprint Backlog
- What was completed?
- What happens to uncompleted items?
- Sprint Goal
- Was it accomplished?
- If not, WHY?
- Product Demo
- Revisit the Product Vision, Product Goal, Roadmap, Release plan
- Any impediments that need help to resolve, which are impacting the team

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Discuss at least three things that do not occur at a Sprint Review.

What do we NOT do during the Sprint Review?

- **Formal sign-off**
- **Presentation of Product Backlog items that do not meet the Definition of Done**
- **Discussion of work hours or budget**

STOP CALLING IT "THE DEMO"

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Identify at least three outcomes for a Sprint Review.

Where do we go from here?

- **Revised Product Backlog**
- **Release the increment**
- **Cancel further development**

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Discuss at least three negative impacts that arise when the Scrum Team disregards one or more of the elements of the Sprint Review.

It's all about solving the customer's problem(s).

- **May not have a full understanding of next steps in creating value**
- **May not be clear on stakeholder's view of the product**
- **Stakeholders may not understand what's going on**
- **Funding might be cut prematurely**
- **Team does not get closure on most recent sprint**

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Sprint Retrospective

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

List the participants of the Sprint Retrospective and describe at least two responsibilities for the Developers, Product Owner, and ScrumMaster during the Sprint Retrospective.

Developers

- Review the results of the previous sprint
- Look for ways to improve their development practices

Product Owner

- Offer observations on the results of the previous sprint
- Participate in retrospective discussions

ScrumMaster

- Ensure that the event takes place
- Maintain the timebox

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Agile Retrospectives

- What worked well?
- What didn't work well?
- What can we do differently next time?





Derby, Esther and Larsen, Diana. 2006. *Agile Retrospectives: Making Good Teams Great. Frisco, TX: Pragmatic Bookshelf*

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

CSM-1.11 Describe at least three possible effects of skipping the Sprint Retrospective



“Well… we’re waiting!”

RESOURCES

(validated 250625)

https://www.funretrospectives.com/

http://www.thiagi.com/
http://www.tastycupcakes.org
http://www.retrospectivewiki.org

(Check Amazon for books on Icebreakers and Party Games)

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Suggested Retrospective Examples to Explore

Sailboat

what moves us forward, what holds us back,
where are we going, what risks are there

Starfish

What should we…
start doing, stop doing, keep doing,
do more of, do less of

GBU

the good, the bad, and the ugly

Good/Great

What was good? What would make it GREAT?

3 L's

Liked, Lacked, Learned

Success Criteria

intention, target, successful if, failure if

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Scrum Artifacts

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Product Backlog

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

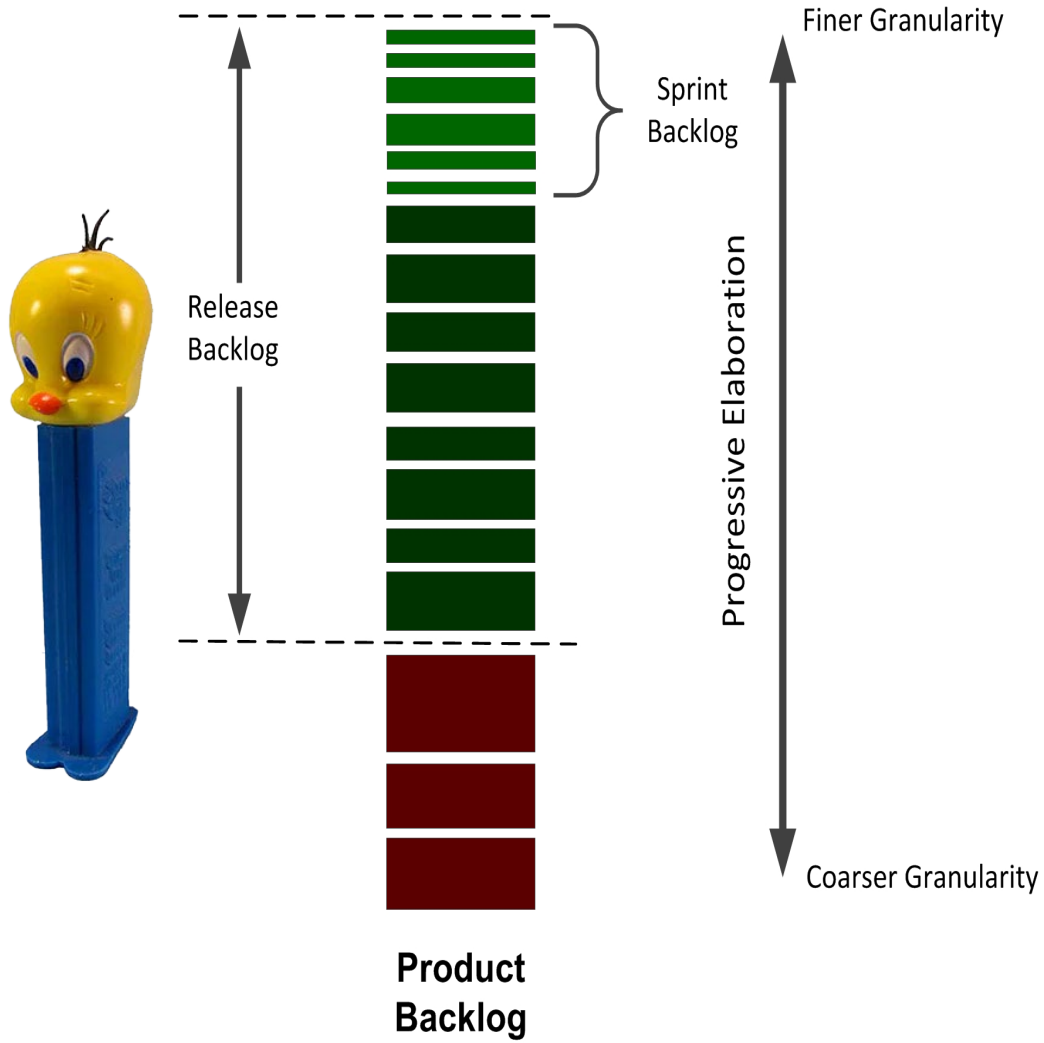
Identify at least three essential characteristics of the Product Backlog.

What is the Product Backlog?

- **Dynamic**
- **Ordered**
- **Having varying levels of detail**

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Product Backlog Visualized



- Product Owner owns and orders the Product Backlog according to the highest valued items at any given moment.

- This can include:
  - Adding Items
  - Removing Items
  - Splitting Items
  - Re-ordering Items

- ANYONE can write and propose Product Backlog Items, but it is the Product Owner who decides where they go on the Product Backlog

- "If it is not on the Product Backlog, it doesn't exist"

  That is, no work should be done by the Development Team without the Product Owners knowledge and consent.

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Top Six Agile Lifecycle Management Tools

(in MY experience)

JIRA
by Atlassian
(commonly used with Confluence, also by Atlassian)

ServiceNow

Rally
by Broadcom

Digital.ai Agility
(formerly VersionOne)

IBM Targetprocess
by Apptio

Azure DevOps
by Microsoft

There are literally 100s of other tools out there.
I recommend Capterra for comparing tools:
www.capterra.com

With the right amount of investment
in time, money, and customization,
they all essentially do the same thing.

Pick ONE and stick with it.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Product Backlog Items

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

List at least four elements of a Product Backlog Item

What is a Backlog Item?

- **Item ID**
- **Description**
- **Size / Complexity**
- **Order**
- **Acceptance Criteria**
- **Dependencies**
- **Parent / Child Relationship**
- **Business Value**

And many others…

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Product Backlog Item Types

A PBI (product backlog item) is first and foremost an increment of value that we intend to deliver to a customer.

Examples

- **Features**
  (increments of value)
- **Enhancement Requests**
- **Production Defect Fixes**
- **Technical Debt**
  (Optional, Unavoidable, Avoidable)

Sometimes people include **Compliance Requirements** as PBIs. MOST compliance concerns should be handled by the DoD criteria. Occasionally, there might be something that is broad, general, and applicable to the entire product vs. pieces that can be updated at the PBI level.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Product Backlog Item Formats/Templates

Examples

Connextra

(aka "user stories")

Job Stories

Agile Lifecycle Management (ALM) Tool Templates

(e.g. JIRA, ADO, Rally, VersionOne, etc.)

These are covered much more extensively in the CSPO course.

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Q: What format should we use for OUR Product Backlog Items?

A: Whatever the Scrum Team agrees on in terms of capturing enough detail to GET TO WORK creating Customer Value ASAP!!

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Sprint Backlog

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Identify at least three essential characteristics of the Sprint Backlog.

What is the Sprint Backlog?

- **Just enough detail**
- **Highly visible**
- **Real-time snapshot of the Developers' work for the sprint**
- **Locked-down, fixed scope that represents a REAL commitment for the Sprint**
- **Aligns with and supports the Sprint Goal**

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Explain how the Sprint Backlog can be changed, how responsibilities for making changes are divided among the Scrum Team members, and the limits of these changes.

What about Commitment?
Tough to aim at a moving target…

- **Developers can add/remove/modify tasks at any time**

- **Product Owner can remove selected Product Backlog items and thus modify the Sprint Backlog indirectly**

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

In Scrum, we NEVER assume.
The conversation never stops.
When in doubt, talk it out.



“You should NEVER assume.

Because when you ASSUME,
you make an ASS of U and ME!”

-Felix Unger, *The Odd Couple*

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

What is included in the Sprint Backlog?

- **Product Backlog Items that have been committed to during Sprint Planning**

- **Any tasks or other information that the Developers feel are necessary to organize and manage themselves during the Sprint**

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

The Increment

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

in•cre•ment
(ˈɪn krə mənt, ˈɪŋ-)

n.
1. something added or gained; addition; increase.
2. the act or process of increasing.
3. an amount by which something increases.
4. one of a series of regular additions: *deposits in increments of $500.*
5.
a. the difference between two values of a variable; a change, positive, negative, or zero, in an independent variable.
b. the increase of a function due to an increase in the independent variable.

[1375–1425; late Middle English < Latin *incrēmentum* an increase]

in`cre•men′tal (-ˈmɛn tl) *adj.*
in`cre•men′tal•ly, *adv.*

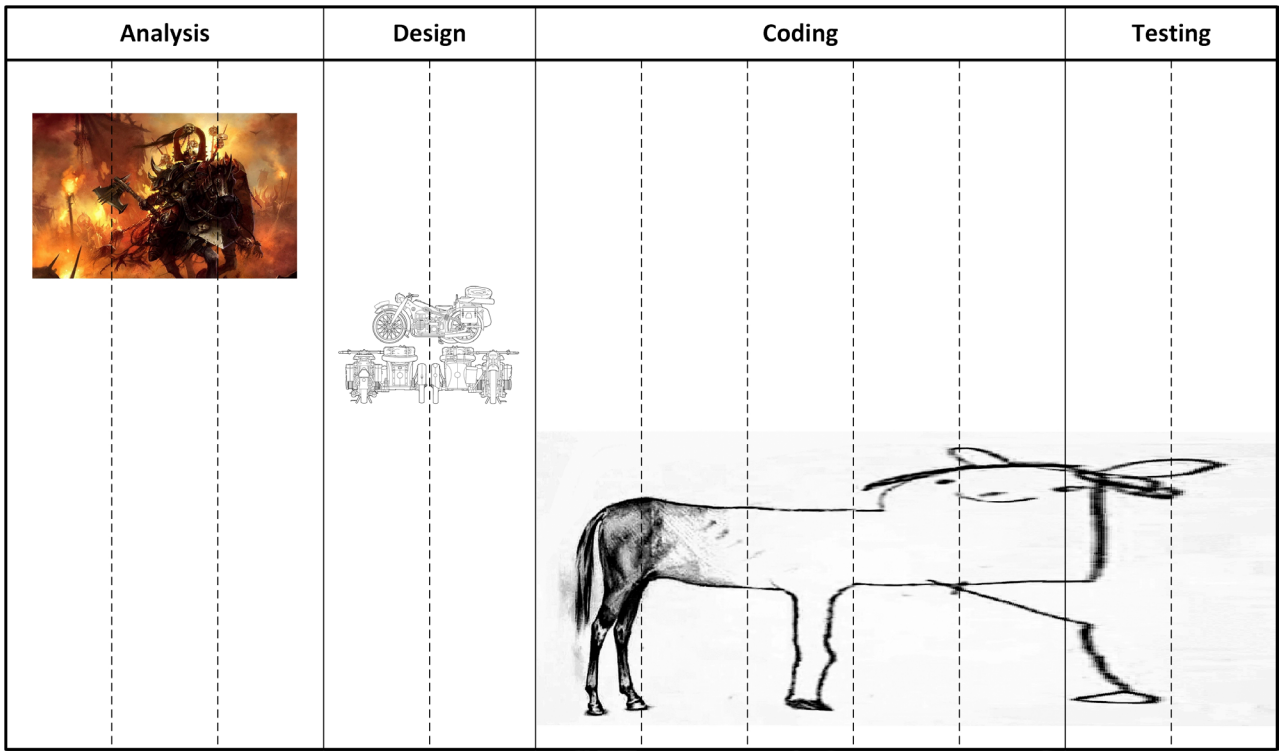
Random House Kernerman Webster's College Dictionary, © 2010 K Dictionaries Ltd. Copyright 2005, 1997, 1991 by Random House, Inc. All rights reserved.

With the product increment, we are adding value to the product; whether it is a single feature, collection of features, a single Sprint's worth of work, or numerous Sprint's worth of work.

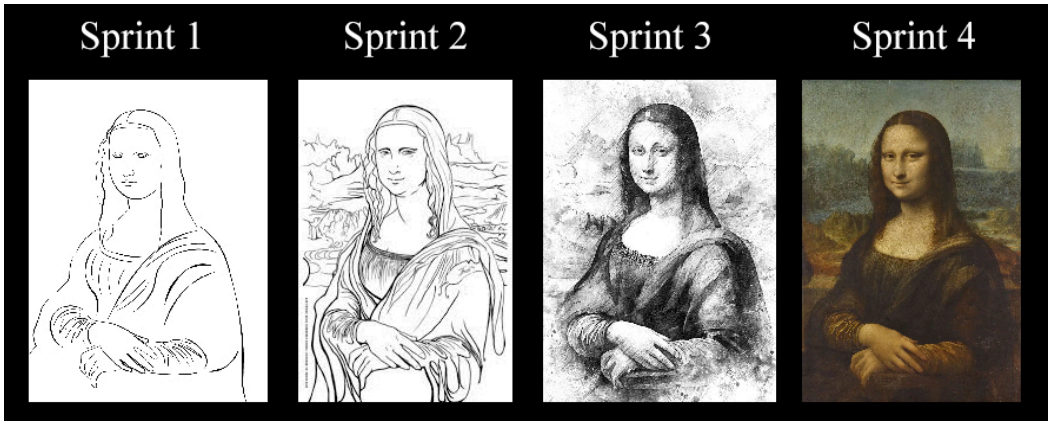
When a PBI meets the DoD, an Increment is born.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Waterfall



Scrum

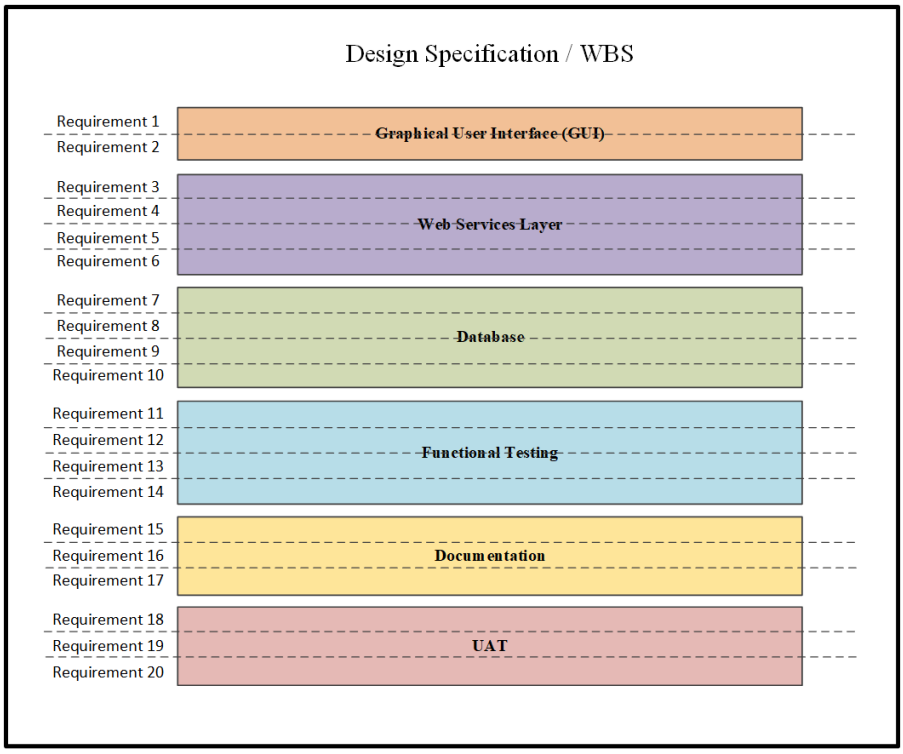


 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

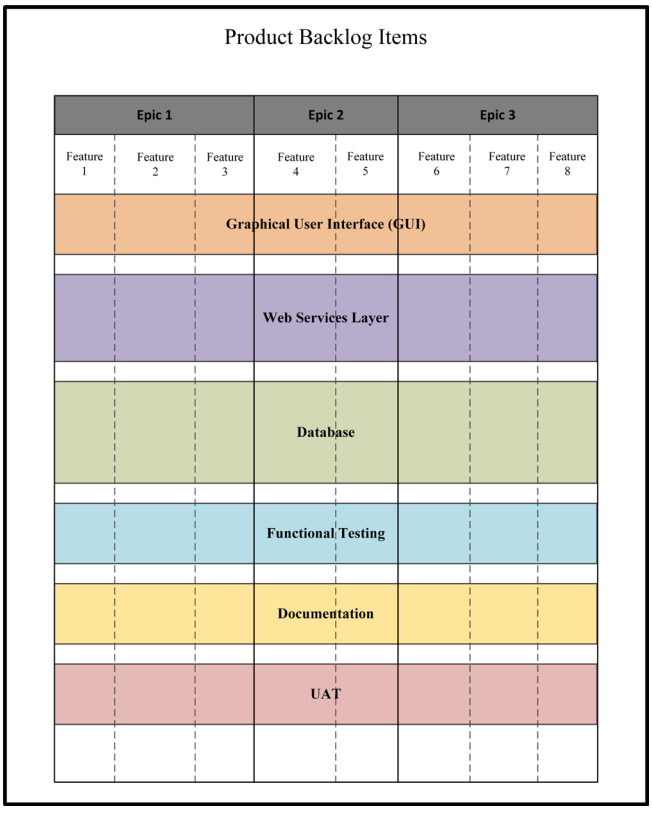
Horizontal Components

Component Teams



Vertical Slices / Increments

Feature Teams



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Definition of Done

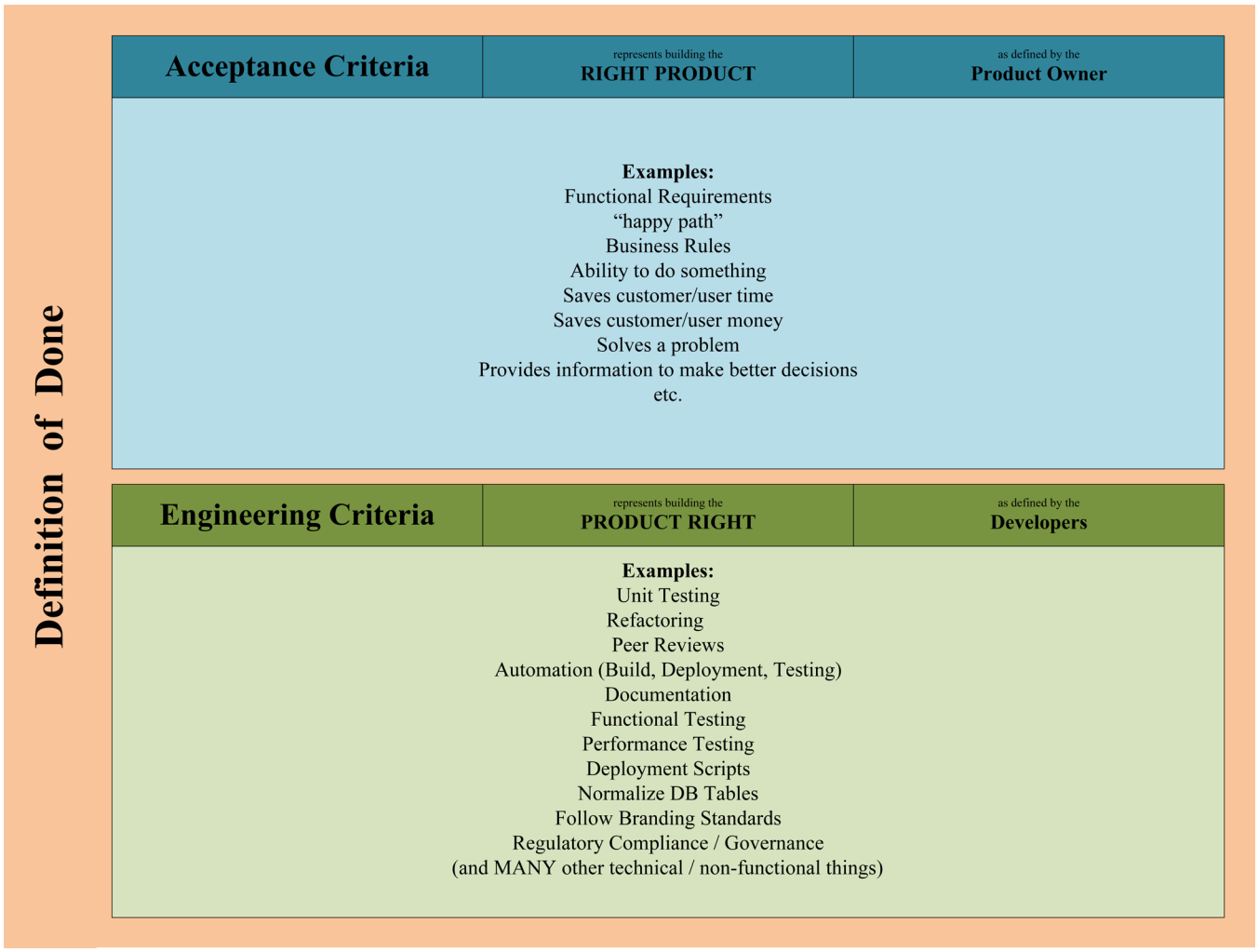
Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-1.15 Explain at least three advantages of
a strong Definition of Done
CSM-1.16 Outline at least one way to create a Definition of Done

Definition of Done

The Definition of Done (DOD) is decided, agreed upon, and updated as needed by each Scrum Team. The DOD includes all tasks and activities necessary to ensure that each Product Backlog Item results in a Potentially Shippable Product Increment.

Two key components of the **Definition of Done** are:



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-3.2 Explain the impact of accumulating technical debt

Technical Debt Quadrant

Martin Fowler
14 October 2009



There's been a few posts over the last couple of months about Technical Debt that's raised the question of what kinds of design flaws should or shouldn't be classified as Technical Debt.

A good example of this is Uncle Bob's post saying a mess is not a debt. His argument is that messy code, produced by people who are ignorant of good design practices, shouldn't be a debt. Technical Debt should be reserved for cases when people have made a considered decision to adopt a design strategy that isn't sustainable in the longer term, but yields a short-term benefit, such as making a release. The point is that the debt yields value sooner but needs to be paid off as soon as possible.

To my mind, the question of whether a design flaw is or isn't debt is the wrong question. Technical Debt is a metaphor, so the real question is whether or not the debt metaphor is helpful about thinking about how to deal with design problems, and how to communicate that thinking. A particular benefit of the debt metaphor is that it's very handy for communicating to non-technical people.

I think that the debt metaphor works well in both cases - the difference is in nature of the debt. A mess is a reckless debt which results in crippling interest payments or a long period of paying down the principal. We have a few projects where we've taken over a code base with a high debt and found the metaphor very useful in discussing with client management how to deal with it.

The debt metaphor reminds us about the choices we can make with design flaws. The prudent debt to reach a release may not be worth paying down if the interest payments are sufficiently small - such as if it were in a rarely touched part of the code-base.

So, the useful distinction isn't between debt or non-debt, but between prudent and reckless debt.

There's another interesting distinction in the example I just outlined. Not just is there a difference between prudent and reckless debt, there's also a difference between deliberate and inadvertent debt. The prudent debt example is deliberate because the team knows they are taking on a debt, and thus puts some thought as to whether the payoff for an earlier release is greater than the costs of paying it off. A team ignorant of design practices is taking on its reckless debt without even realizing how much hock it's getting into.

Reckless debt may not be inadvertent. A team may know about good design practices, even be capable of practicing them, but decide to go "quick and dirty" because they think they can't afford the time required to write clean code. I agree with Uncle Bob that this is usually a reckless debt, because people underestimate where the Design Payoff Line is. The whole point of good design and clean code is to

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

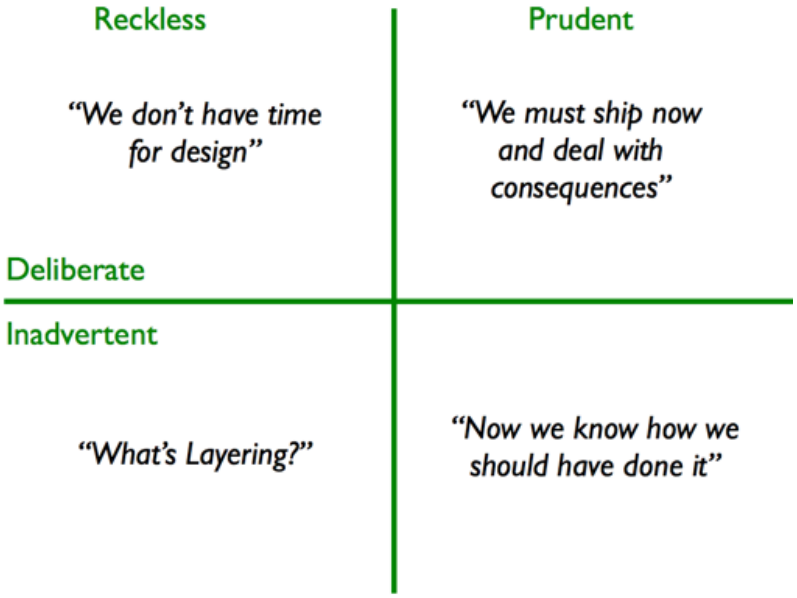
make you go faster - if it didn't people like Uncle Bob, Kent Beck, and Ward Cunningham wouldn't be spending time talking about it.

Dividing debt into reckless/prudent and deliberate/inadvertent implies a quadrant, and I've only discussed three cells. So, is there such a thing as prudent-inadvertent debt? Although such a thing sounds odd, I believe that it is - and it's not just common but inevitable for teams that are excellent designers.

I was chatting with a colleague recently about a project he'd just rolled off from. The project that delivered valuable software, the client was happy, and the code was clean. But he wasn't happy with the code. He felt the team had done a good job, but now they realize what the design ought to have been.

I hear this all the time from the best developers. The point is that while you're programming, you are learning. It's often the case that it can take a year of programming on a project before you understand what the best design approach should have been. Perhaps one should plan projects to spend a year building a system that you throw away and rebuild, as Fred Brooks suggested, but that's a tricky plan to sell. Instead what you find is that the moment you realize what the design should have been, you also realize that you have an inadvertent debt. This is the kind of debt that Ward talked about in his video.

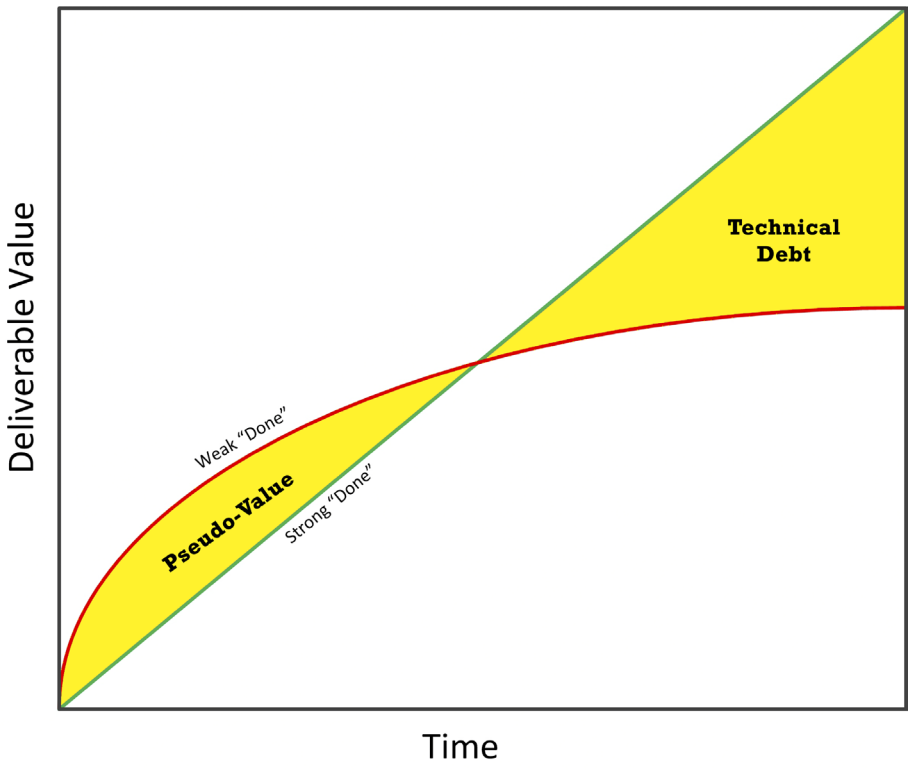
The decision of paying the interest versus paying down the principal still applies, so the metaphor is still helpful for this case. However, a problem with using the debt metaphor for this is that I can't conceive of a parallel with taking on a prudent-inadvertent financial debt. As a result, I would think it would be difficult to explain to managers why this debt appeared. My view is this kind of debt is inevitable and thus should be expected. Even the best teams will have debt to deal with as a project goes on - even more reason not to recklessly overload it with crummy code.



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Technical Debt

When the DOD is "weak" there is an initial perception of a "quick win" as the feature is marked complete and deployed to the appropriate environment. This weak definition may be lacking in automated testing, documentation, or other key factors eventually resulting in Technical Debt.



Source: Ron Jeffries

Accumulation of technical debt is a sign of practices that are irresponsible, immature, disingenuous, lacking in big picture thinking, near-sighted, and purely reactive (vs. responsive).

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting



In Aikido, we seek to bring harmony and balance to every interaction, taking negative energy and redirecting it in a positive way.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Agile Facilitation

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-2.1 Describe at least three situations in which the Scrum Master could serve the needs of the Scrum Team or organization through facilitation

The Product Owner and Developers are arguing about estimates

The Developers are arguing among themselves regarding a potential solution

The Product Owner and Developers get together for a brainstorming session on new feature ideas based on customer feedback

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-2.2 Demonstrate at least three techniques for facilitating group decision making

Let's decide something…

- **Dot voting**
- **Monopoly voting**
- **Thumb voting**
- **Constellations**
- **Tribes**
- **Role playing**
- **Remember the Future**
- **Choose your own adventure**

RESOURCES
(validated 250625)

https://www.funretrospectives.com/

http://www.thiagi.com/
http://www.tastycupcakes.org
http://www.retrospectivewiki.org

(Check Amazon for books on Icebreakers and Party Games)

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Facilitation Tools

miro

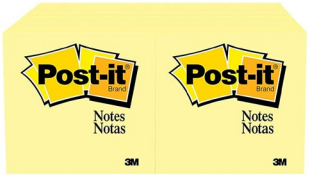


MURAL

























 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Agile Coaching

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-2.3 Discuss how facilitating, teaching, mentoring, and coaching are different

Agile Coaching Competency Framework

Agile-Lean Practitioner Ability to learn and deeply understand Agile frameworks and Lean principles, not only at the level of practices, but also at the level of the principles and values that underlie the practices enabling appropriate application as well as innovation.

Professional Coaching Ability to act as a coach, with the client's interest determining the direction, rather than the coach's expertise or opinion.

Facilitating Neutral process holder that guides the individual's, team's, or organization's process of discovery, holding to their purpose and definition of success.

Mentoring Ability to impart one's experience, knowledge and guidance to help grow another in the same or similar knowledge domains.

Teaching Ability to offer the right knowledge, at the right time, taught in the right way, so that individuals, teams and organizations metabolize the knowledge for their best benefit.

Technical Mastery Ability to get your hands dirty architecting, designing, coding, test engineering, or performing some other technical practice, with a focus on promoting technical craftsmanship through example and teaching-by-doing. And, expertise in agile scaling patterns or structures.

Business Mastery Ability to apply business strategy and management frameworks to employ agile as a competitive business advantage such as Lean Start-Up, product innovation techniques, flow-based business process management approaches, and other techniques that relate to innovating in the business domain.

Transformation Mastery Ability to facilitate, catalyze and (as appropriate) lead organizational change and transformation. This area draws on change management, organization culture, organization development, systems thinking, and other behavioral sciences.



Source: Lyssa Adkins (2010), Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Coaching frameworks

GROW
G for Goal setting: define the short- and long-term goals
R for Reality: explore the current situation
O for Options: identify and evaluate different action strategies
W for Will: what will you do by when?

FUEL
F for frame the Conversation
U for Understand the Current State
E for Explore the Desired State
L for Lay out a Success Plan

OSKAR
O for Outcome or objective: start with want your goal from the conversation
S for Scale: What is the measurement of where the person is towards achieving their goal (1 to 10)
K for Know-how: review the skills, actions and potential for your team member to move towards 10
A for Affirm + Action: Affirm what is working well and commit to the actions to move forward.
R for Review: Determine the success of the past OSKAR steps and how to move forward.

CLEAR
C is for Contracting: exploring the goal of the coaching conversation
L is for Listening: active listening to learn about the situation
E is for Exploring: reviewing all options and possibilities of the situation
A is for Action: choosing a path and next steps.
R is for Review: Exploring how to conversation went

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Service to the Scrum Team

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

"Managers"
(aka bad bosses)













 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Ten Characteristics of a Servant-Leader

Listening

Empathy

Healing

Awareness

Persuasion

Conceptualization

Foresight

Stewardship

Commitment to the growth of people

Building community

Source: Greenleaf, Robert K. (2002). Servant Leadership: A Journey into The Nature of Legitimate Power and Greatness. New Jersey: Paulist Press.

Source: Larry C. Spears (1991), Ten Characteristics of a Servant Leader

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-3.1 Describe three scenarios where the Scrum Master acts as a leader for the Scrum Team

The Scrum Master ensures that everyone understands the Agile Values and Principles and how they relate to the Scrum practices

The Scrum Master practices the Socratic method to problem solving so that the Scrum Team formulates its own solutions to problems

The Scrum Master establishes a book-of-the-month club within the team wherein they read various Agile related books and discuss how they might learn and grow from these

The Scrum Master plans a team building event where the Scrum Team members can get to know each other more closely and establish better working relationships

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-3.7 Apply at least one technique that could help resolve an impediment
CSM-3.6 Discuss at least two ways that the Scrum Master assists the Scrum Team with impediments

Makes impediments visible

Work directly with the Scrum Team to understand and resolve impediments using brainstorming and facilitation

Establishes Communities of Practice (COP)

Quantify impediments in terms of impact to ROI (i.e. cost) and present to executive leadership

Build connections, trust, influence, and power to get things done across the organization

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Share some GOOD servant leader vibes…

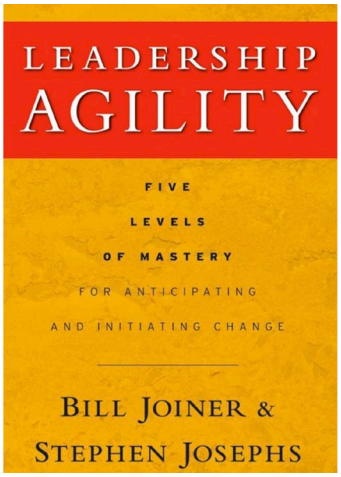


"People will forget what you said, people will forget what you did, but people will never forget how you make them feel."
- Maya Angelou

Servant leaders make people feel valued and respected.

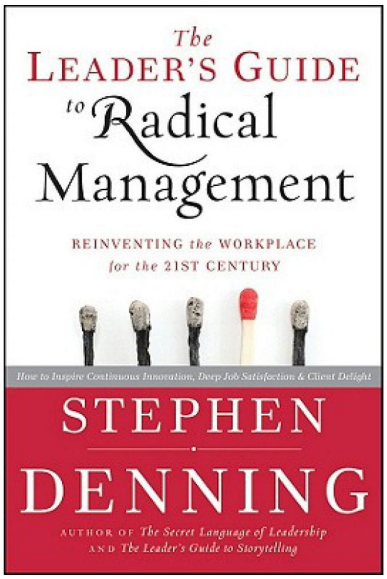
Be a servant leader.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



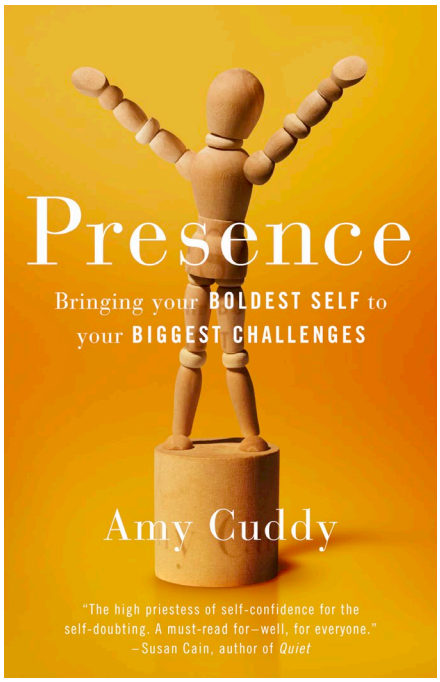


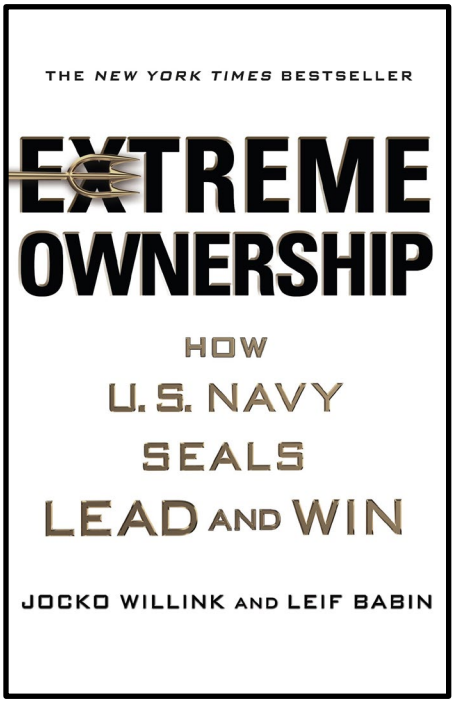


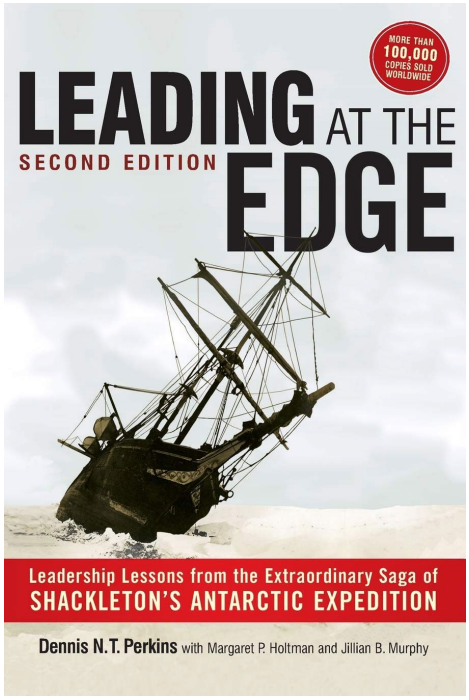


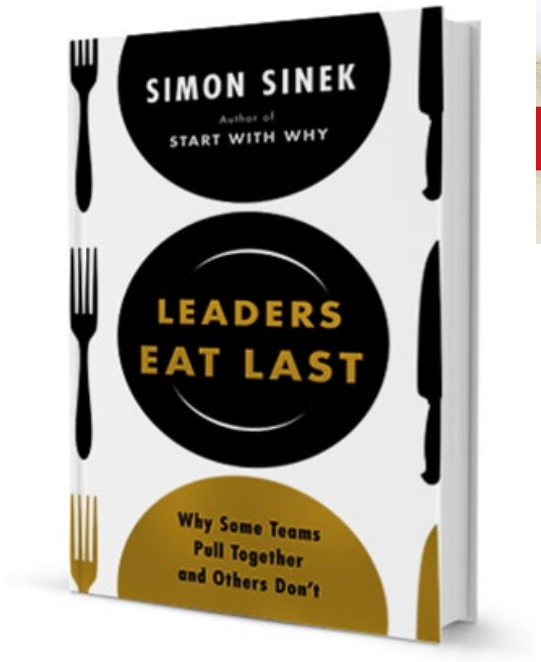
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

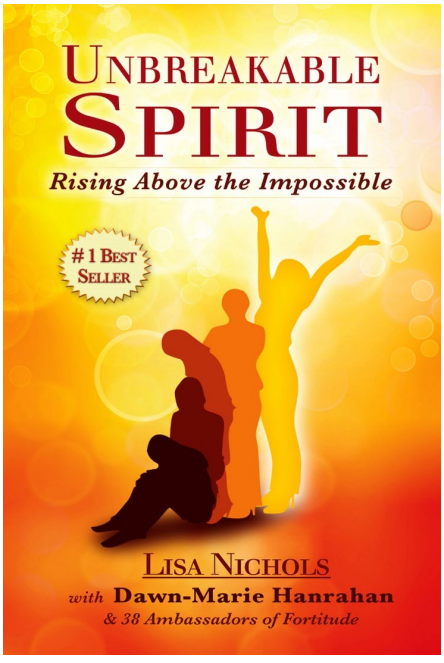
5 Must-Read Books on **Leadership**











 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

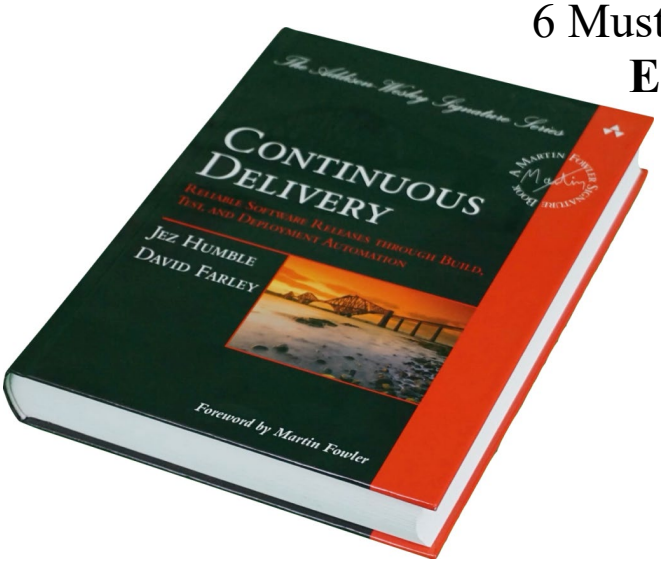
CSM-3.3 List at least three development practices that could help their Scrum Team deliver a high-quality Increment and reduce technical debt each Sprint

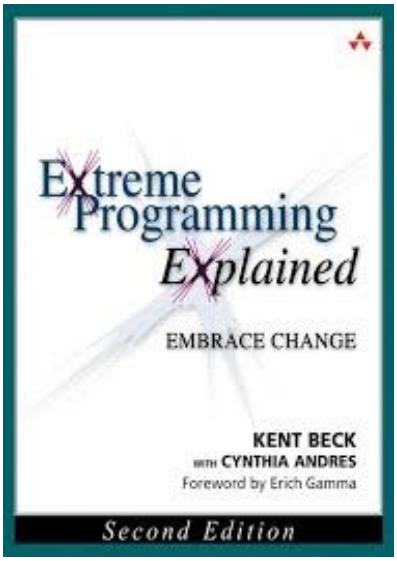
Extreme Programming (XP)

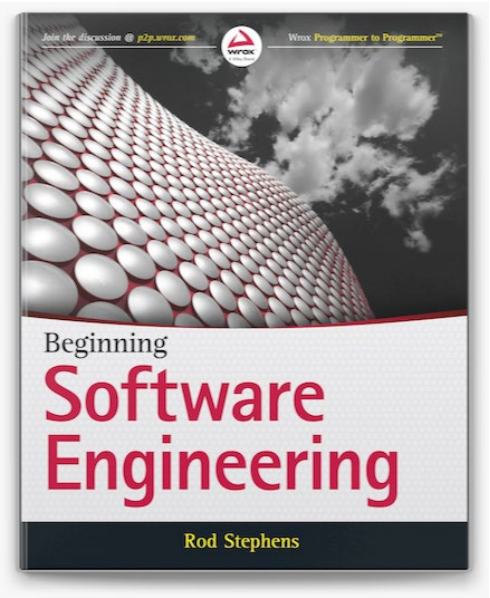
- **Test-driven development**
- **Pair Programming**
- **Continuous Integration**
- **Collective Code Ownership**
- **Refactoring**

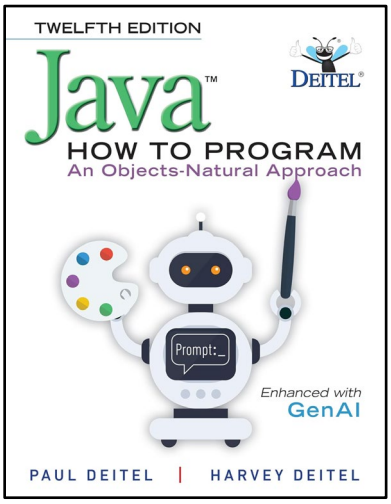
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

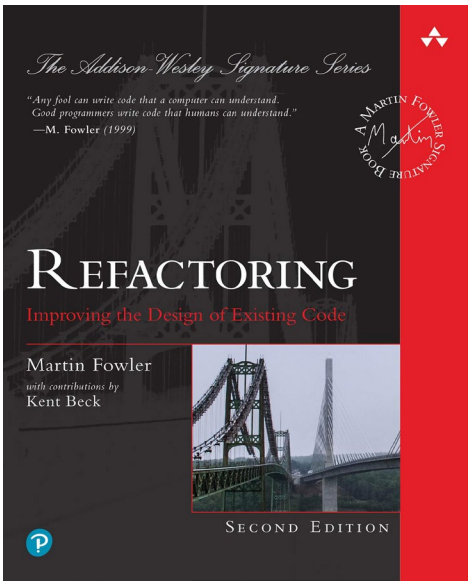
6 Must-Read Books on **Engineering**

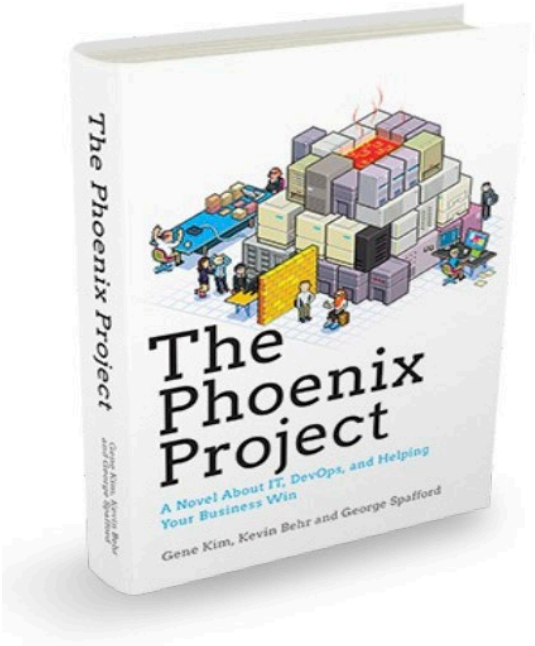












 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Daniel's 52 Book Challenge

Cancel Netflix, etc.
(~$100-300/month value)

Instead, buy 4-5 books / month
from Amazon

Or, buy a subscription to:
Addison-Wesley / Pearson
O'Reilly
Wiley

Or, look for books here:
Hoopla
Project Guttenberg
Libby

THEN…

Read a book / week for 1 year
selected from the bibliography
in the back of this workbook

Be amazed at your new level of genius!!
👐 💡 👐 💡 👐 💡 👐

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Other ways to close the gap in your lack of technical knowledge:

Khan Academy
https://www.khanacademy.org/computing

CodeCademy

Udemy

Pluralsight

Grasshopper

YouTube videos

Coding Bootcamps
e.g. Flatiron Academy

KnowledgeHut Courses
Java Development

Take the
Certified Scrum Developer (CSD)
course

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Service to the Product Owner

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-3.4 Explain at least three ways the Scrum Master could support the Product Owner

Helping find techniques for effective Product Goal definition and Product Backlog management

Helping the Scrum Team understand the need for clear and concise Product Backlog items

Helping establish empirical product planning for a complex environment

Facilitating stakeholder collaboration as requested or needed

Explaining technical matters and estimates that the Developers have provided when the Product Owner is unclear

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

"Failure **is** *an option here. If things are not failing, you are not innovating enough."*

-Elon Musk



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Take the CSPO course!!

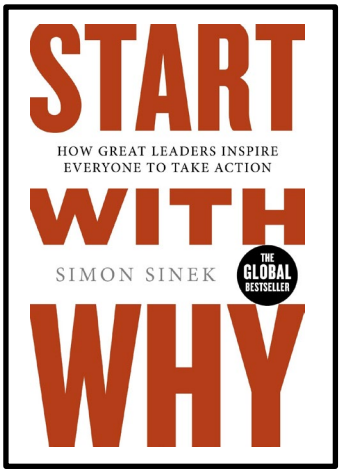
The CSPO Learning Objectives fall into the following categories:

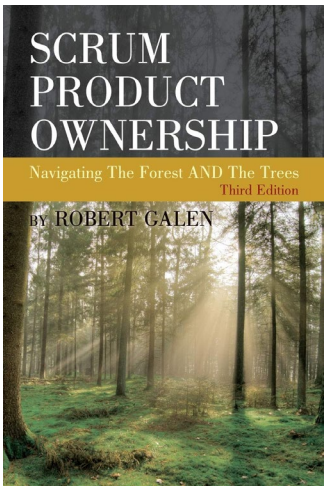
1. **Product Owner Core Competencies**
2. **Goal Setting and Planning**
3. **Understanding Customers and Users**
4. **Validating Product Assumptions**
5. **Working with the Product Backlog**

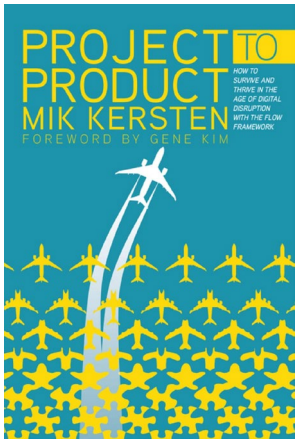
NOTE: *You don't have to take* **MY** *CSPO course. Just take SOMEBODY's CSPO course or you may have a hard time coaching Product Owners. You will also get 16 SEUs for your CSM renewal and another 16 PDUs if you have the PMP.*

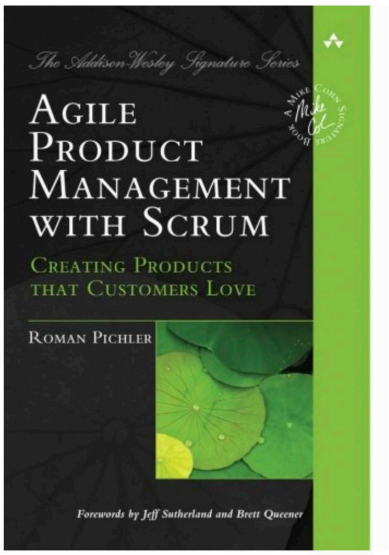
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

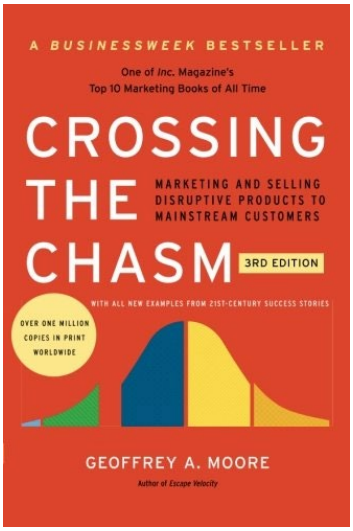
9 Must-Read Books on **Product Ownership**

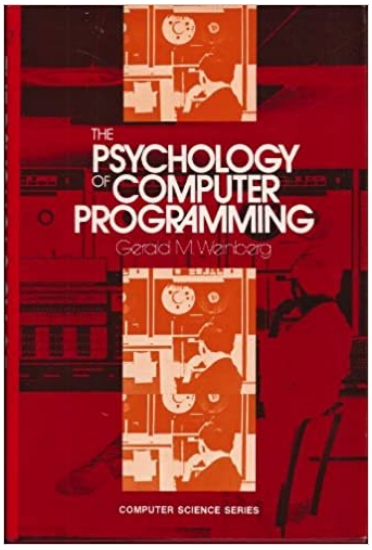


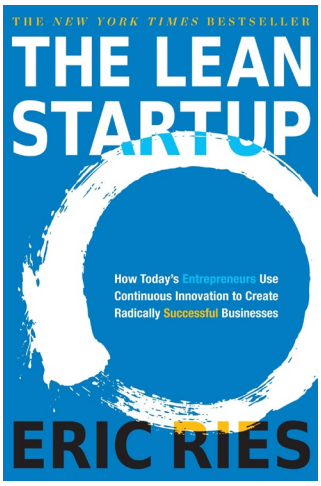


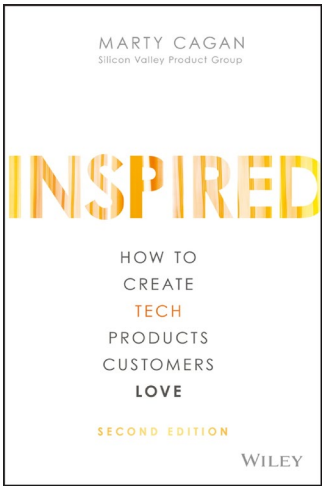


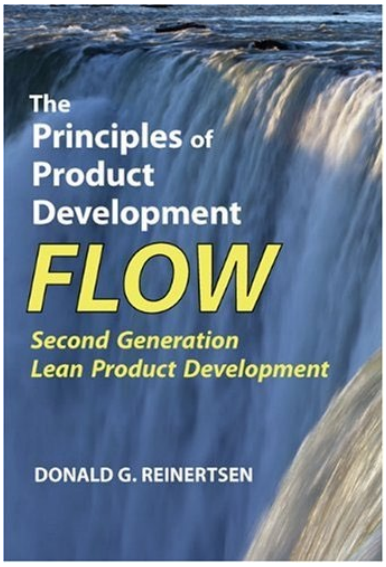












 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Service to the Organization

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-3.8 Summarize at least one organizational design change caused by adopting Scrum

Challenges

- Geographical distribution
- People in multiple project teams
- Harmful incentive and HR policies
- No constructive safe-to-fail culture
- Over-allocation of resources
- Unrealistic timelines
- Lack of vision and strategy
- Insurmountable technical debt

Solutions

- Eliminate single-function groups; hire cross-functional people and full-stack roles
- Establish non-traditional career paths
- Re-think annual appraisals
- Slow down to go faster
- Be focused and intentional
- Recognize creative ways to meet compliance requirements
- Spend more time DELIVERING value and less time TALKING about delivering value

It is worthwhile to do a **SWOT** analysis for your team to identify factors that will impact your team.
(Strengths, Weaknesses, Opportunities, Threats)

Your Enterprise Agile Coach needs your help and is there to help you.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Leadership Behavior

Constructive

- Management is not telling people what to do
  People closest to the work make the majority of the decisions
  Understanding that autonomy instills trust and consequently better workmanship from the team.
- Acknowledging that only one of the constraints in the “iron triangle” can be FIXED (commited), one may be FIRM (targeted), and the last one FLEXIBLE (unlimited)
- The more flexible the third constraint, the more probable the second constraint will be.
- Providing respectful, constructive feedback on the product

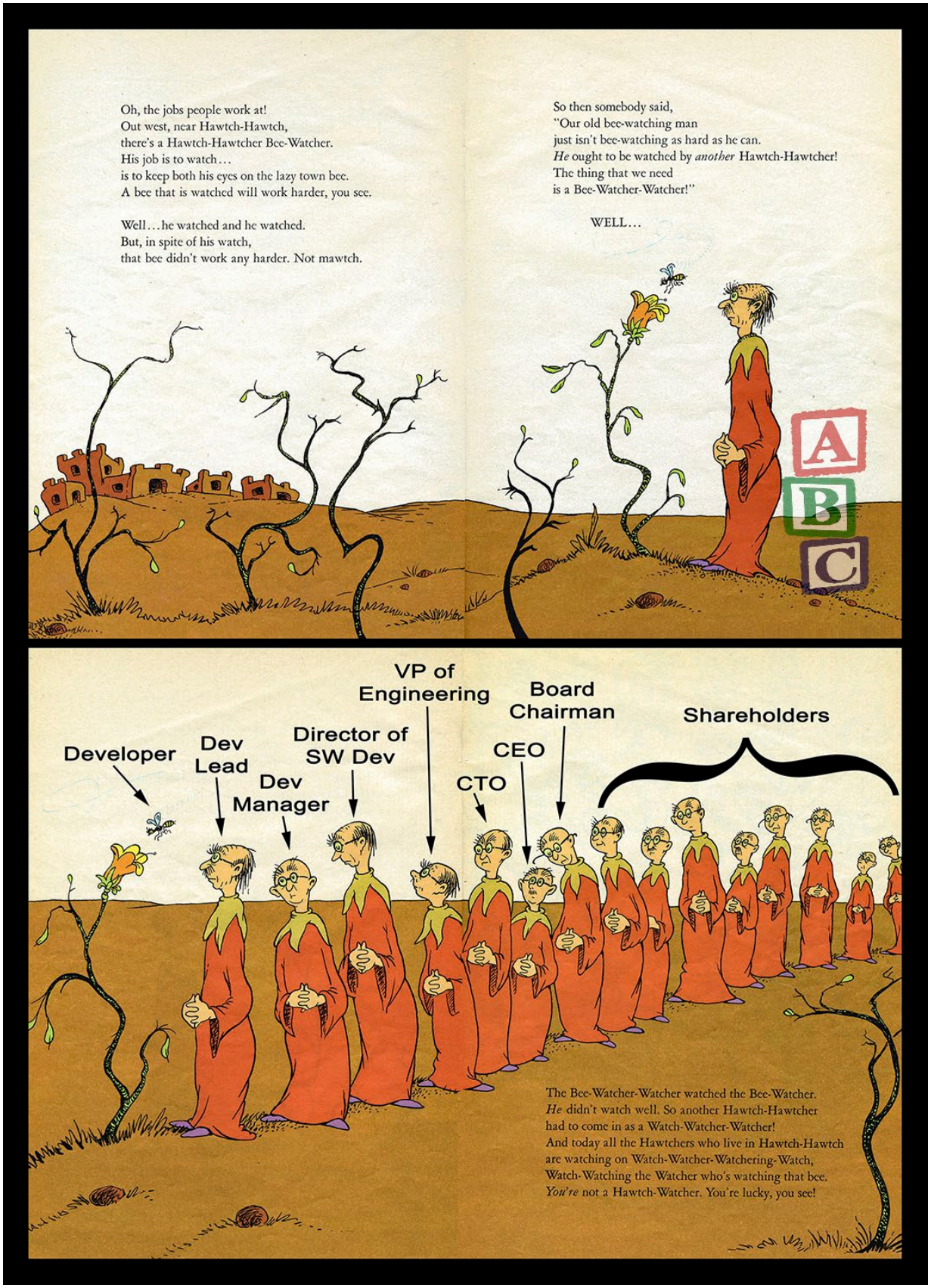
Destructive

- Micromanaging the team by expecting excessive documentation of time, tasks, etc.
- Unrealistic expectations around the triple constraint
- Expecting people to be on multiple teams or acting as PO / SM for multiple teams
- Not adopting an Agile mindset
- Ignoring the dysfunctions that Scrum makes transparent about the organization
- Throwing tantrums like an over-grown toddler

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

More workers, less managers

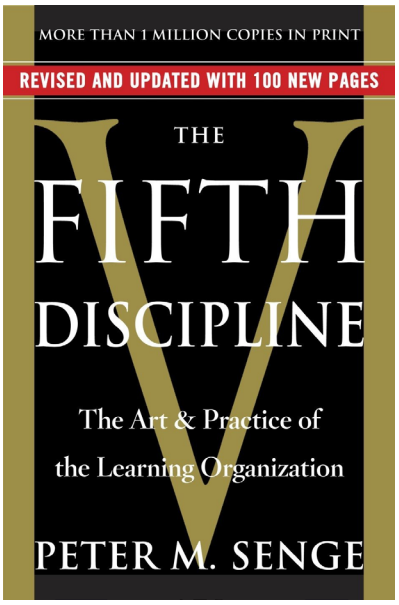
"Does this management model make us look fat?"

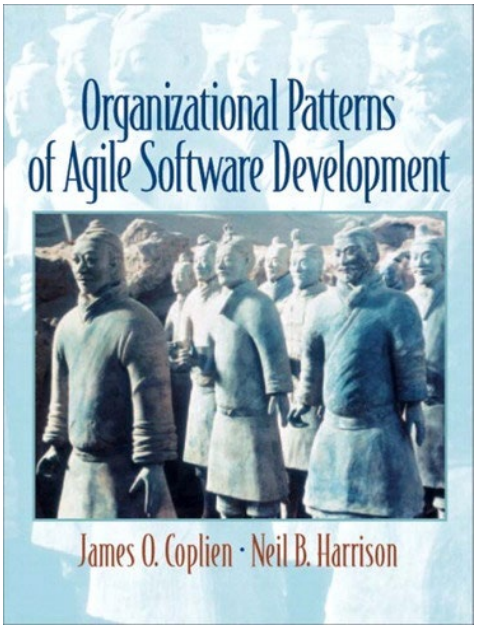


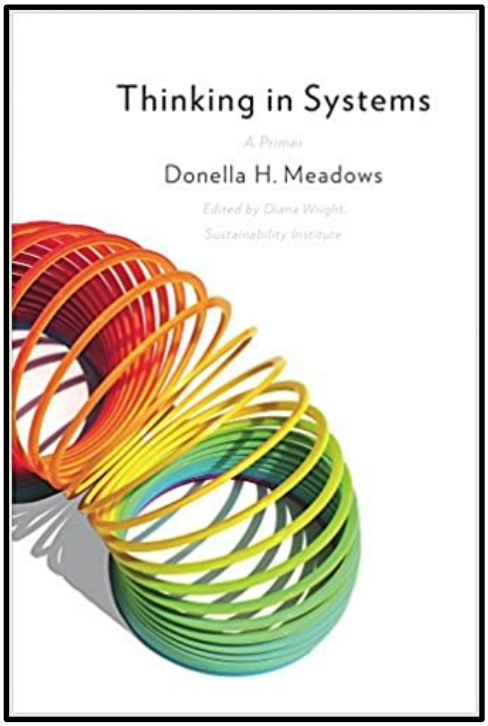
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

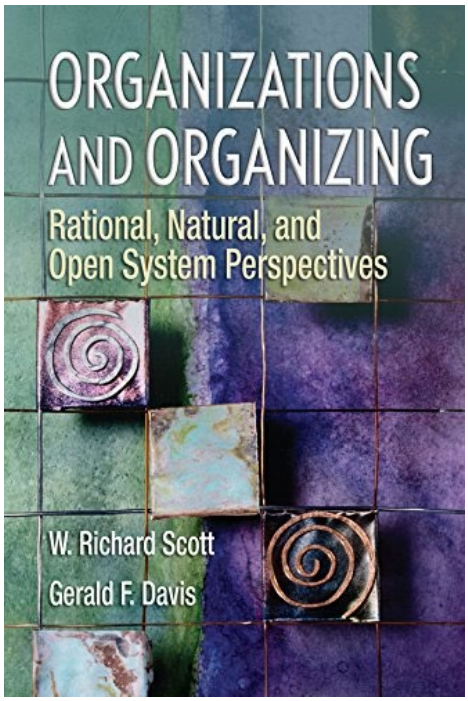
5 Must-Read Books on **Organization Development and Change**











 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Bonus / Extra Credit:

Organizational Change Models

Lewin

The McKinsey 7-S model

Kotter

ADKAR

Nudge theory

Bridges transition model

Kübler-Ross

Satir

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

CSM-3.5 Describe at least three organizational impediments that can affect Scrum Teams

Scrum Is Hard and Disruptive

Ken Schwaber, 2006

1. Scrum is a framework for iterative, incremental development using cross-functional, self-managing teams. It is built on industry best practices, lean thinking, and empirical process control.

2. Scrum is optimized for high yield product management and product development. Scrum is particularly appropriate for high risk, complex, large projects and can be used when other parts of the endeavor are hardware or even waterfall development.

3. If waterfall suits current needs, continue using it.

4. An enterprise can use Scrum as a tool to become the best product development and management organization in its market. Scrum will highlight every deficiency and impediment that the enterprise has so the enterprise can fix them and change into such an organization.

5. Whenever an enterprise modifies or only partially implements Scrum, it is hiding or obscuring one or more dysfunctionalities that restrict its competence in product development and management.

6. The iterative, incremental nature of Scrum puts stress on the product development organization to improve its engineering skills and on the product management organization to optimize the return on investment of every release and project. The phrase, “That can’t be done here” really means that it will be very difficult to do so. The gap between current practices and target practices is a measure of incompetence and competitive risk.

7. The use of Scrum to become an optimized product development and management organization is a change process that must be led from the top and requires change by everyone within the enterprise. Change is extremely difficult and fraught with conflict and may take many years of sustained effort. Turnover of staff and management can be expected.

8. The most serious impediments to using Scrum are habits of waterfall, predictive thinking over the last twenty to thirty years; these have spawned command and control management, belief that demanding something will make it happen, and the willingness of development to cut quality to meet dates. These are inbred habits that we aren’t even aware of anymore.

9. The focus of using Scrum is the change from old habits to new ways of doing business. Scrum is not implemented or rolled-out as a process; it is used to foment change.

10. Scrum is not a methodology that needs enhancing. That is how we got into trouble in the first place, thinking that the problem was not having a perfect methodology. Effort centers on the changes in the enterprise that is needed.

11. Iterative, incremental development is much harder than waterfall development; everything that was hard in waterfall engineering practices now has to be done every iteration, and this is incredibly hard. It is not impossible but has to be worked toward over time.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

12. Managing a release or project to deliver only the highest value functionality and not deliver the rest optimizes value [and] is the job of product management and customers.

13. Self-managing teams are extremely productive. When they work closely with the customer to derive the best solution to a need, they and the customer are even more productive.

14. A team consists of people under pressure to do their best. Conflict is natural, and the team needs to know how to deal with the conflict and have resources to draw on when needed.

15. The role of an enterprises management changes from telling people what to do to leading and helping everyone do their best to achieve goals. People aren't resources and managers aren't bosses.



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

We Tried Baseball and It Didn't Work

Ron Jeffries, 2006

An allegory? Sarcasm? Humorous pastiche? You decide.

The fanatical proponents of baseball tell us that it is a very exciting game, fun to play and fun to watch. They are clearly either stupid or evil or both, because we tried baseball and it didn't work.

First of all, the requirements for the game are stupid: it does not scale. They say you need at least nine players on a side. That's stupidly inefficient. The minimum number of players is clearly four: three men on and one batting. That's how we played: four people on a side.

With only four players, we didn't need all those bases (another obvious inefficiency and lack of scaling), so we just used one base: home. When a batter makes a hit, he just runs around a square (but see below), coming back to the one "home" base. At first, we had the runner run around the full ninety-foot square designated by the baseball proponents, but for some reason the baseball guys couldn't see that 360 feet is too far for the runner to run on most hits. No one ever got all the way around without being put out. We found a circle with diameter of 28.64 feet to be nearly ideal. (The circle, with its optimal ratio of area to circumference, is much more efficient than the square proposed by the baseball pushers.)

The equipment "required" by these baseball guys is ludicrous, anachronistic, discriminating, and cruel. First of all, the "baseball" is supposed to be covered in horsehide. This is simultaneously cruel to horses, and discriminatory against the traditional providers of ball covering, pigs and cows. In view of this, we used other balls. We tried an uncovered ball of twine, but its continually decreasing size made the game difficult, especially in later innings. Golf balls went too far and were hard to catch. One of our teams used the heads of dolls they had borrowed from their sisters, but this led to difficulties at home. How the proponents of baseball could push a game that leads to such problems is beyond us.

We finally settled on woolen socks, stuffed with scrap paper and knotted. These were easy to catch, did not go too far, and the dangling end of the sock makes them easy to throw.

The so-called "bats" are also a bad idea. First of all, they are nothing like those furry insect-catchers of the night-time sky. The resulting name confusion has delayed many a game. In addition, "baseball bats" (we had to go to a fully-qualified name) are expensive to purchase and difficult to make. We tried letting each batter use any implement he

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

wanted, but this delayed the game and made statistics keeping (a key part of baseball according to the proponents) difficult. We settled on using sticks from around the field. Pine works best, as it is easy to break off the trees, and as the hands get sticky from the pine tar, it becomes easier to catch the sock, when you are in the field.

Speaking of the field, another problem that baseball proponents have not properly handled is that with one person pitching, one catching, and only two in the field, the ball is too often hit to where the fielders aren't, resulting in too much running and also in games that go on too long and have scores that are far too high. We finally figured out – why the original baseball guys didn't, we can't imagine – that using one player as catcher is horribly inefficient. Instead, if the brancher (the term "batter" is both ambiguous and inappropriate in view of the equipment changes we found necessary to make the game work at all) misses a pitch, he merely picks up the sock himself and tosses it back to the pitcher. This lets you have all four defensive players in the field, which is much more efficient in use of resources, another proof that baseball doesn't scale.

The thing that finally condemns the entire "baseball" idea, however is this: even with all these improvements, the game is no fun at all.

We tried baseball, and it didn't work.



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum is like putting your organization on a diet/fitness plan.

Most people fail at Agile transformation for the same reasons that most people fail at losing weight and getting in shape.

They want ALL the benefits while making NONE of the sacrifices and they lack the rigor and discipline to commit to doing what it takes to be successful.

Stop cheating and making excuses.

Get that organization in shape… NOW!



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum Simulation

CSM-1.8 Perform a Sprint Planning
CSM-1.9 Perform a Sprint Review
CSM-1.10 Perform a Sprint Retrospective
CSM-2.2 Demonstrate at least three techniques for facilitating group decision making

Practicing Scrum
For this exercise, we will use Scrum, as we have learned it, to build a brochure for a product or service. As a suggestion, you could create a brochure which teaches the Agile Manifesto values, principles, and Scrum Practices or promotes CSM / CSPO courses.
This exercise is NOT optional.

E.G.
Your team creates a brochure that we can use as marketing material for our product or service, whatever that might be.

I have found that this simulation very closely follows real world behaviors, discoveries, patterns, etc. that people will encounter at in their organizations. It's also a lot of fun…

Release Planning (30 minutes)
Each team must have a Product Owner, ScrumMaster, and Developers.

Product Owner will define the vision and strategy for the brochure as well as the Product Backlog.

The **Developers** will define how these are implemented with the resources at hand; i.e. understanding MURAL and how to create things, where to find resources, etc.

The **ScrumMaster** will ensure that things are running smoothly and that everyone has what they need by establishing burndown charts, working agreements, facilitating the events, capturing a task board, retrospective feedback, etc.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum Simulation Schedule

Prep time	30

	Minutes
Sprint Planning Meeting	5
Daily Scrum - Day One	3
Day One	10
Daily Scrum - Day Two	3
Day Two	10
Daily Scrum - Day Three	3
Day Three	10
Sprint Review	5
Sprint Retrospective	5
Total	54

Afternoon Break	30

	Minutes
Sprint Planning Meeting	5
Daily Scrum - Day One	3
Day One	10
Daily Scrum - Day Two	3
Day Two	10
Daily Scrum - Day Three	3
Day Three	10
Sprint Review	5
Sprint Retrospective	5
Total	54

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Afterword

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



Dear student-

You just sat through my class and became certified.

I am very passionate about coaching and training. I go to great lengths and do everything in my power to bring a quality, enjoyable classroom experience to my students. Mostly, I try to have fun and spread the fun with everyone. In making these classes fun, I like to kid around and such. Hopefully you picked up on my sense of humor and sarcasm. If not, see the article on the next page from Seth Godin…

I also push the envelope and do not treat you with kid gloves. I love you too much as my students to sugarcoat the message for you. Agile is a HUGE paradigm shift in thinking. Improvement is a change. Don't expect to be Agile without making changes in your thinking.

Because of the diverse background and experience of the classroom participants, it is not always possible to discuss what is important to each person. If you need additional help, I am happy to provide whatever coaching I can, free of charge, beyond the classroom. So, please reach out and I will see what I can do for you. In some cases, I can answer your questions quickly via Email. In other cases, we can figure out what makes sense in terms of me coming to coach your organization.

Also, feedback is extremely important. It is a key driving principle of Agile and especially Scrum, as we have shown. We ask each student to complete a survey for the class in order to capture feedback on how we might improve and to provide one last opportunity to discuss things that were enjoyable or things that were a disappointment. Please take the time to let us know how we did. As it says in the Terms and Conditions, please reach out to me first before you talk to anyone else if you weren't happy. Let's understand each other. It's a key to not only Agile, but civilized society.

Even more candidly, if there was something you weren't happy with, have the decency to come to me first and discuss. I can't fix the situation if I don't know there is a problem.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

In some very rare cases (only a handful of cases in >11 years), some people have complained to the Scrum Alliance about something that was essentially a customer satisfaction issue. That is completely disrespectful and contrary to the Scrum Values to do that. It tells me that the person didn't learn a thing about being Agile and should relinquish their certification…

Be human. Be real. Let's talk. Give me a chance to make things right and delight you.

I wish you all the best in your Agile journey.

Peace and blessings.

-Daniel

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



Most valuable career advice EVER:

"Just be yourself, sir. Whatever happens, they can't take that away from you."

-Coleman, *Trading Places*

Also, some advice that has helped me over the years from one of the best leadership books ever written…

"Whatever you do, work at it with all your heart, as working for the Lord, not for human masters."

-Colossians 3:23-

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

For the **one person** who didn't get the joke…

The fabled comedian is killing it at a club that seats 400. One guy in the back, though, isn't laughing.

Miles Davis was shunned by a few people in the audience, even at his coolest.

The theater critic at the Times might not like this play, the one that made people cry and sold tickets for years.

And just about every blog post and book listing collects a trolling comment from someone who didn't like it, didn't read it or didn't agree with it (or all three) and isn't shy about speaking up with a sharp tongue.

For those people, the message from the creator of the work is clear:

"It's not for you."

Unanimity is impossible unless you are willing to be invisible. We can be unanimous in our lack of feedback for the invisible one.

For everyone else, though, the ability to say, "It's not for you," is the foundation for creating something brave and important. You can't do your best work if you're always trying to touch the untouchable, or entertain those that refuse to be entertained.



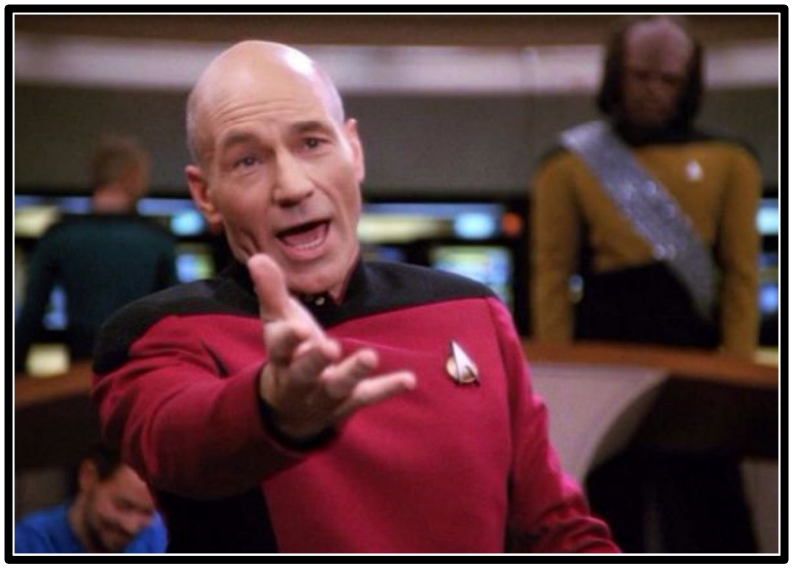
"It's not for you."

This is easy to say and incredibly difficult to do.

You don't have much choice, though, not if you want your work to matter.

Seth Godin
February 01, 2013

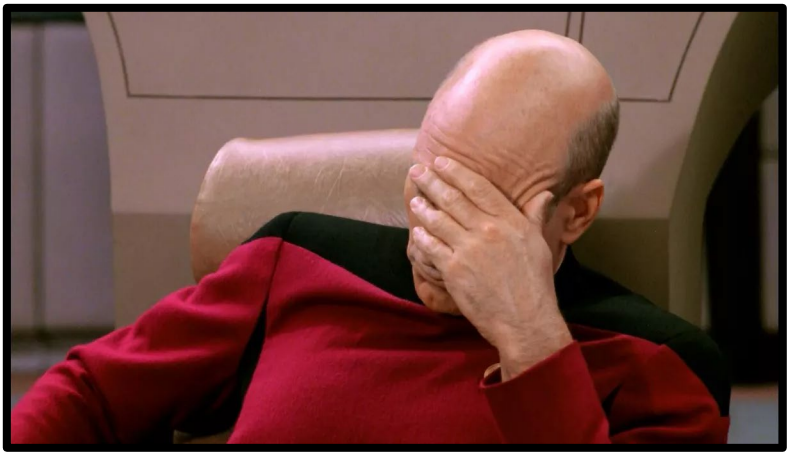
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved



"An important aspect of Scrum is being able to figure out things for yourself. That is a an absolutely **CRITICAL** concept to grasp."



"That makes perfect sense. How do we do that? What are the steps??"



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

What's Next?

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

1. **PLEASE… go back and RE-READ EVERYTHING AGAIN.** All of your questions have been answered already in this workbook and by Daniel in the class. Maybe watch the videos again on the KnowledgeHut site Prism. Download and read the Preclass Work PDF as well. (see **Leading Yourself** in this workook)
2. Daniel will check the Zoom connection logs to make sure you were in class the FULL time.
3. Assuming you were there the full time AND participating in the course with video camera turned on, etc. Daniel will contact KnowledgeHut to inform them that you can be uploaded to the Scrum Alliance system. At this point, if you need help or support, you must contact KnowledgeHut.
4. You will receive an Email from Scrum Alliance with additional instructions in it. Read these instructions very carefully. If you do NOT receive an Email from Scrum Alliance within 3 days of the course, you must contact KnowledgeHut. Please do not contact me. I have no access to KnowledgeHut systems. I also have no access to Scrum Alliance systems. If you need help please reach out to those organizations respectively.
5. Once you do receive the Email from Scrum Alliance, if you need help or support you must contact Scrum Alliance. Neither Daniel nor KnowledgeHut can do anything for you at this point.
6. In general, Scrum Alliance will ask you to create a profile by entering additional personal information.
7. You will have a dashboard in your profile, which is where you manage your certification. Your certificate is here. Your digital badge and logos are here. You log your SEUs here. The link for the CSM exam is here.
8. You will take the exam (and pass it on the first try!!)
9. You will accept your CSM license agreement.
10. You will go out and be an awesome Scrum Master in the world as a newly certified CSM.
11. Read this article:
    Getting Hired With Your New Certification
    https://resources.scrumalliance.org/Article/put-your-new-scrummaster-skills-to-work
12. I don't do resume reviews. I also don't have any magical words of wisdom when it comes to finding your first job as a CSM other than this: BUILD A PERSONAL BRAND ONLINE
    If you put in the sweat equity, you will eventually be successful. It may take time. Have faith.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Appendix

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Purpose of the Scrum Guide

We developed Scrum in the early 1990s. We wrote the first version of the Scrum Guide in 2010 to help people worldwide understand Scrum. We have evolved the Guide since then through small, functional updates. Together, we stand behind it.

The Scrum Guide contains the definition of Scrum. Each element of the framework serves a specific purpose that is essential to the overall value and results realized with Scrum. Changing the core design or ideas of Scrum, leaving out elements, or not following the rules of Scrum, covers up problems and limits the benefits of Scrum, potentially even rendering it useless.

We follow the growing use of Scrum within an ever-growing complex world. We are humbled to see Scrum being adopted in many domains holding essentially complex work, beyond software product development where Scrum has its roots. As Scrum's use spreads, developers, researchers, analysts, scientists, and other specialists do the work. We use the word "developers" in Scrum not to exclude, but to simplify. If you get value from Scrum, consider yourself included.

As Scrum is being used, patterns, processes, and insights that fit the Scrum framework as described in this document, may be found, applied and devised. Their description is beyond the purpose of the Scrum Guide because they are context sensitive and differ widely between Scrum uses. Such tactics for using within the Scrum framework vary widely and are described elsewhere.

Ken Schwaber & Jeff Sutherland November 2020

© 2020 Ken Schwaber and Jeff Sutherland

This publication is offered for license under the Attribution Share-Alike license of Creative Commons, accessible at https://creativecommons.org/licenses/by-sa/4.0/legalcode and also described in summary form at https://creativecommons.org/licenses/by-sa/4.0/. By utilizing this Scrum Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Purpose of the Scrum Guide 1
Scrum Definition 3
Scrum Theory 3
Transparency 3
Inspection 4
Adaptation 4
Scrum Values 4
Scrum Team 5
Developers 5
Product Owner 5
Scrum Master 6
Scrum Events 7
The Sprint 7
Sprint Planning 8
Daily Scrum 9
Sprint Review 9
Sprint Retrospective 10
Scrum Artifacts 10
Product Backlog 10
Commitment: Product Goal 11
Sprint Backlog 11
Commitment: Sprint Goal 11
Increment 11
Commitment: Definition of Done 12
End Note 13
Acknowledgements 13
People 13
Scrum Guide History 13

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum Definition

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

A Product Owner orders the work for a complex problem into a Product Backlog.
The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
Repeat

Scrum is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. **The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory. Scrum is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.**

Various processes, techniques and methods can be employed within the framework. Scrum wraps around existing practices or renders them unnecessary. Scrum makes visible the relative efficacy of current management, environment, and work techniques, so that improvements can be made.

Scrum Theory

Scrum is founded on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials.

Scrum employs an iterative, incremental approach to optimize predictability and to control risk. Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed.

Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint. These events work because they implement the empirical Scrum pillars of transparency, inspection, and adaptation.

Transparency

The emergent process and work must be visible to those performing the work as well as those receiving the work. With Scrum, important decisions are based on the perceived state of its three formal artifacts. Artifacts that have low transparency can lead to decisions that diminish value and increase risk.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Transparency enables inspection. Inspection without transparency is misleading and wasteful.

Inspection

The Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems. To help with inspection, Scrum provides cadence in the form of its five events.

Inspection enables adaptation. Inspection without adaptation is considered pointless. Scrum events are designed to provoke change.

Adaptation

If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted. The adjustment must be made as soon as possible to minimize further deviation.

Adaptation becomes more difficult when the people involved are not empowered or self-managing. A Scrum Team is expected to adapt the moment it learns anything new through inspection.

Scrum Values

Successful use of Scrum depends on people becoming more proficient in living five values:
Commitment, Focus, Openness, Respect, and Courage
The Scrum Team commits to achieving its goals and to supporting each other. Their primary focus is on the work of the Sprint to make the best possible progress toward these goals. The Scrum Team and its stakeholders are open about the work and the challenges. Scrum Team members respect each other to be capable, independent people, and are respected as such by the people with whom they work. The Scrum Team members have the courage to do the right thing, to work on tough problems.

These values give direction to the Scrum Team with regard to their work, actions, and behavior. The decisions that are made, the steps taken, and the way Scrum is used should reinforce these values, not diminish or undermine them. The Scrum Team members learn and explore the values as they work with the Scrum events and artifacts. When these values are embodied by the Scrum Team and the people they work with, the empirical Scrum pillars of transparency, inspection, and adaptation come to life building trust.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Scrum Team

The fundamental unit of Scrum is a small team of people, a Scrum Team. The Scrum Team consists of one Scrum Master, one Product Owner, and Developers. Within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focused on one objective at a time, the Product Goal.

Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint. They are also self-managing, meaning they internally decide who does what, when, and how.

The Scrum Team is small enough to remain nimble and large enough to complete significant work within a Sprint, typically 10 or fewer people. In general, we have found that smaller teams communicate better and are more productive. If Scrum Teams become too large, they should consider reorganizing into multiple cohesive Scrum Teams, each focused on the same product. Therefore, they should share the same Product Goal, Product Backlog, and Product Owner.

The Scrum Team is responsible for all product-related activities from stakeholder collaboration, verification, maintenance, operation, experimentation, research and development, and anything else that might be required. They are structured and empowered by the organization to manage their own work. Working in Sprints at a sustainable pace improves the Scrum Team's focus and consistency.

The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint. Scrum defines three specific accountabilities within the Scrum Team: the Developers, the Product Owner, and the Scrum Master.

Developers

Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.

The specific skills needed by the Developers are often broad and will vary with the domain of work. However, the Developers are always accountable for:

Creating a plan for the Sprint, the Sprint Backlog;
Instilling quality by adhering to a Definition of Done;
Adapting their plan each day toward the Sprint Goal; and,
Holding each other accountable as professionals.

Product Owner

The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

The Product Owner is also accountable for effective Product Backlog management, which includes:

Developing and explicitly communicating the Product Goal;
Creating and clearly communicating Product Backlog items;
Ordering Product Backlog items; and,
Ensuring that the Product Backlog is transparent, visible and understood.

The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable.

For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review.

The Product Owner is one person, not a committee. The Product Owner may represent the needs of many stakeholders in the Product Backlog. Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

Scrum Master

The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide. They do this by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organization.

The Scrum Master is accountable for the Scrum Team's effectiveness. They do this by enabling the Scrum Team to improve its practices, within the Scrum framework.

Scrum Masters are true leaders who serve the Scrum Team and the larger organization. The **Scrum Master serves the Scrum Team** in several ways, including:
Coaching the team members in self-management and cross-functionality;
Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
Causing the removal of impediments to the Scrum Team's progress; and,
Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.

The **Scrum Master serves the Product Owner** in several ways, including:

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Helping find techniques for effective Product Goal definition and Product Backlog management;
Helping the Scrum Team understand the need for clear and concise Product Backlog items;
Helping establish empirical product planning for a complex environment; and,
Facilitating stakeholder collaboration as requested or needed.

The **Scrum Master serves the organization** in several ways, including:

Leading, training, and coaching the organization in its Scrum adoption;
Planning and advising Scrum implementations within the organization;
Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
Removing barriers between stakeholders and Scrum Teams.

Scrum Events

The Sprint is a container for all other events. Each event in Scrum is a formal opportunity to inspect and adapt Scrum artifacts. These events are specifically designed to enable the transparency required. Failure to operate any events as prescribed results in lost opportunities to inspect and adapt. Events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. Optimally, all events are held at the same time and place to reduce complexity.

The Sprint

Sprints are the heartbeat of Scrum, where ideas are turned into value.

They are fixed length events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint.

All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

During the Sprint:

No changes are made that would endanger the Sprint Goal;
Quality does not decrease;
The Product Backlog is refined as needed; and,
Scope may be clarified and renegotiated with the Product Owner as more is learned.

Sprints enable predictability by ensuring inspection and adaptation of progress toward a Product Goal at least every calendar month. When a Sprint's horizon is too long the Sprint Goal may become invalid, complexity may rise, and risk may increase. Shorter Sprints can be employed to generate more learning

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

cycles and limit risk of cost and effort to a smaller time frame. Each Sprint may be considered a short project.

Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows. While proven useful, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision making.

A Sprint could be cancelled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint.

Sprint Planning

Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. This resulting plan is created by the collaborative work of the entire Scrum Team.

The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal. The Scrum Team may also invite other people to attend Sprint Planning to provide advice.

Sprint Planning addresses the following topics:

Topic One: Why is this Sprint valuable?
The Product Owner proposes how the product could increase its value and utility in the current Sprint. The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders. The Sprint Goal must be finalized prior to the end of Sprint Planning.

Topic Two: What can be Done this Sprint?
Through discussion with the Product Owner, the Developers select items from the Product Backlog to include in the current Sprint. The Scrum Team may refine these items during this process, which increases understanding and confidence.

Selecting how much can be completed within a Sprint may be challenging. However, the more the Developers know about their past performance, their upcoming capacity, and their Definition of Done, the more confident they will be in their Sprint forecasts.

Topic Three: How will the chosen work get done?
For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the Developers. No one else tells them how to turn Product Backlog items into Increments of value.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

The Sprint Goal, the Product Backlog items selected for the Sprint, plus the plan for delivering them are together referred to as the Sprint Backlog.

Sprint Planning is timeboxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Daily Scrum

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

The Developers can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management.

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

The Daily Scrum is not the only time Developers are allowed to adjust their plan. They often meet throughout the day for more detailed discussions about adapting or re-planning the rest of the Sprint's work.

Sprint Review

The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations. The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed.

During the event, the Scrum Team and stakeholders review what was accomplished in the Sprint and what has changed in their environment. Based on this information, attendees collaborate on what to do next. The Product Backlog may also be adjusted to meet new opportunities. The Sprint Review is a working session and the Scrum Team should avoid limiting it to a presentation.

The Sprint Review is the second to last event of the Sprint and is timeboxed to a maximum of four hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Sprint Retrospective

The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.

The Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done. Inspected elements often vary with the domain of work. Assumptions that led them astray are identified and their origins explored. The Scrum Team discusses what went well during the Sprint, what problems it encountered, and how those problems were (or were not) solved.

The Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible. They may even be added to the Sprint Backlog for the next Sprint.

The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Scrum Artifacts

Scrum's artifacts represent work or value. They are designed to maximize transparency of key information. Thus, everyone inspecting them has the same basis for adaptation.

Each artifact contains a commitment to ensure it provides information that enhances transparency and focus against which progress can be measured:

For the Product Backlog it is the Product Goal.
For the Sprint Backlog it is the Sprint Goal.
For the Increment it is the Definition of Done.

These commitments exist to reinforce empiricism and the Scrum values for the Scrum Team and their stakeholders.

Product Backlog

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team.

Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining activities. Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items. This is an ongoing activity to add details, such as a description, order, and size. Attributes often vary with the domain of work.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

The Developers who will be doing the work are responsible for the sizing. The Product Owner may influence the Developers by helping them understand and select trade-offs.

Commitment: Product Goal

The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against. The Product Goal is in the Product Backlog. The rest of the Product Backlog emerges to define "what" will fulfill the Product Goal.

A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users or customers. A product could be a service, a physical product, or something more abstract.

The Product Goal is the long-term objective for the Scrum Team. They must fulfill (or abandon) one objective before taking on the next.

Sprint Backlog

The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how).

The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal. Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.

Commitment: Sprint Goal

The Sprint Goal is the single objective for the Sprint. Although the Sprint Goal is a commitment by the Developers, it provides flexibility in terms of the exact work needed to achieve it. The Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives.

The Sprint Goal is created during the Sprint Planning event and then added to the Sprint Backlog. As the Developers work during the Sprint, they keep the Sprint Goal in mind. If the work turns out to be different than they expected, they collaborate with the Product Owner to negotiate the scope of the Sprint Backlog within the Sprint without affecting the Sprint Goal.

Increment

An Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. In order to provide value, the Increment must be usable.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Sprint. The Sprint Review should never be considered a gate to releasing value.

Work cannot be considered part of an Increment unless it meets the Definition of Done.

Commitment: Definition of Done

The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

The moment a Product Backlog item meets the Definition of Done, an Increment is born.

The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment. If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review. Instead, it returns to the Product Backlog for future consideration.

If the Definition of Done for an increment is part of the standards of the organization, all Scrum Teams must follow it as a minimum. If it is not an organizational standard, the Scrum Team must create a Definition of Done appropriate for the product.

The Developers are required to conform to the Definition of Done. If there are multiple Scrum Teams working together on a product, they must mutually define and comply with the same Definition of Done.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

End Note

Scrum is free and offered in this Guide. The Scrum framework, as outlined herein, is immutable. While implementing only parts of Scrum is possible, the result is not Scrum. Scrum exists only in its entirety and functions well as a container for other techniques, methodologies, and practices.

Acknowledgements

People

Of the thousands of people who have contributed to Scrum, we should single out those who were instrumental at the start: Jeff Sutherland worked with Jeff McKenna and John Scumniotales, and Ken Schwaber worked with Mike Smith and Chris Martin, and all of them worked together. Many others contributed in the ensuing years and without their help Scrum would not be refined as it is today.

Scrum Guide History

Ken Schwaber and Jeff Sutherland first co-presented Scrum at the OOPSLA Conference in 1995. It essentially documented the learning that Ken and Jeff gained over the previous few years and made public the first formal definition of Scrum.

The Scrum Guide documents Scrum as developed, evolved, and sustained for 30-plus years by Jeff Sutherland and Ken Schwaber. Other sources provide patterns, processes, and insights that complement the Scrum framework. These may increase productivity, value, creativity, and satisfaction with the results.

The complete history of Scrum is described elsewhere. To honor the first places where it was tried and proven, we recognize Individual Inc., Newspage, Fidelity Investments, and IDX (now GE Medical).

© 2020 Ken Schwaber and Jeff Sutherland

This publication is offered for license under the Attribution Share-Alike license of Creative Commons, accessible at https://creativecommons.org/licenses/by-sa/4.0/legalcode and also described in summary form at https://creativecommons.org/licenses/by-sa/4.0/. By utilizing this Scrum Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

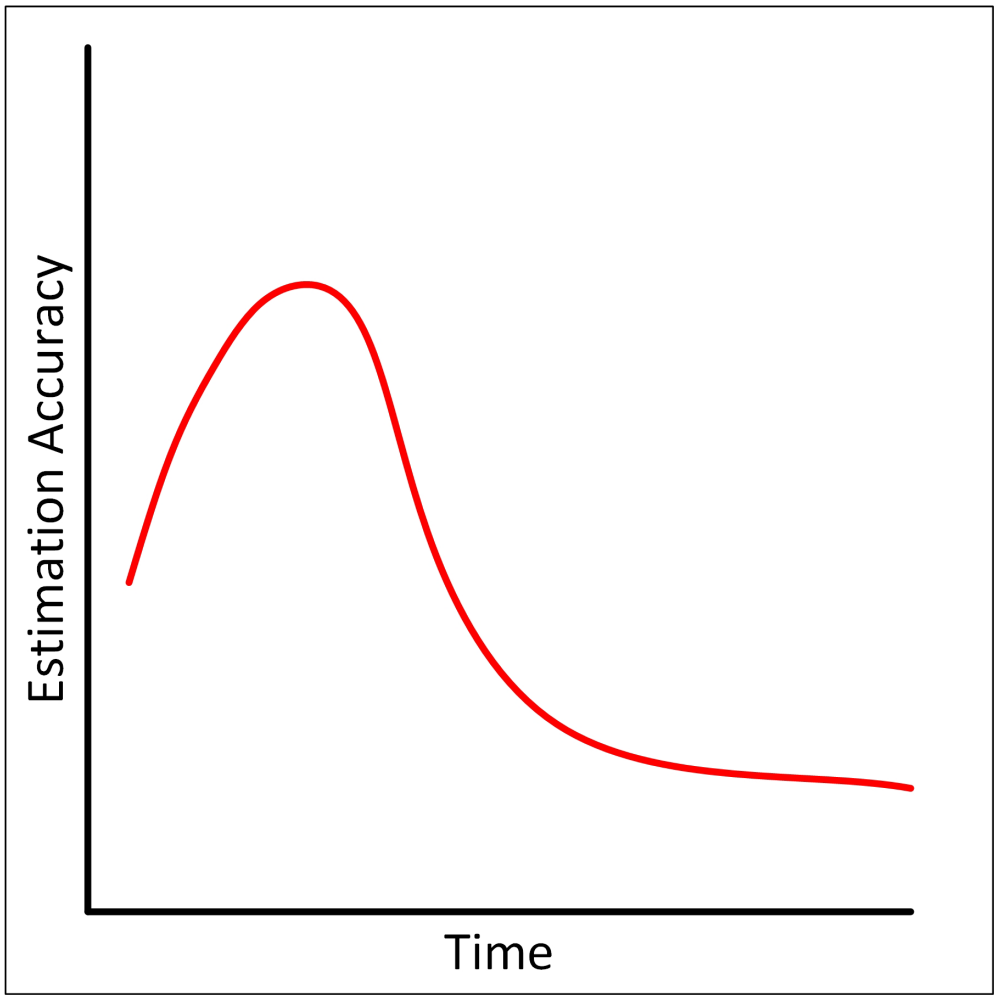
 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Professional Guessing

(aka Estimation)

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Time Spent Providing Estimates



It makes sense to spend a modicum of time formulating an estimate. However, make it quick because the goal isn't to become a good guesser, but rather, to deliver value ASAFP.

There is a diminishing return on time invested for estimation. Accuracy actually gets WORSE the more time you spend. There is a whole body of scholarly research around this topic.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Estimation Methods

Absolute Estimation

Trying to predict or guess the actual value of some unknown, actual characteristic of an item; such as cost or time. Humans are very poor at judging the EXACT value. Also, once some statement is made about cost or time, no matter how tentative or speculative, stakeholders tend to interpret that statement as a commitment. The goal of absolute estimating is attempting to get as close as possible to the actual value.



Person A: "What's the size of that building?"
Person B: "Um, probably 1500 feet."
Person A: "Nope. It's 1454 feet."

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Relative Estimation

Comparing items in relation to one another using some unknown characteristic to get a sense of their size or ranking. We may not know the exact complexity of a feature. However, we are clear that the feature is more complex or less complex than other things we are working on. Overall, we are very good at making comparisons; not so great at guessing at exact values.



Person A: "What are the sizes of these buildings?"
Person B: "That building is taller than this building. That one is taller than both of the others… etc."
Person A: "Yes, that is correct."

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

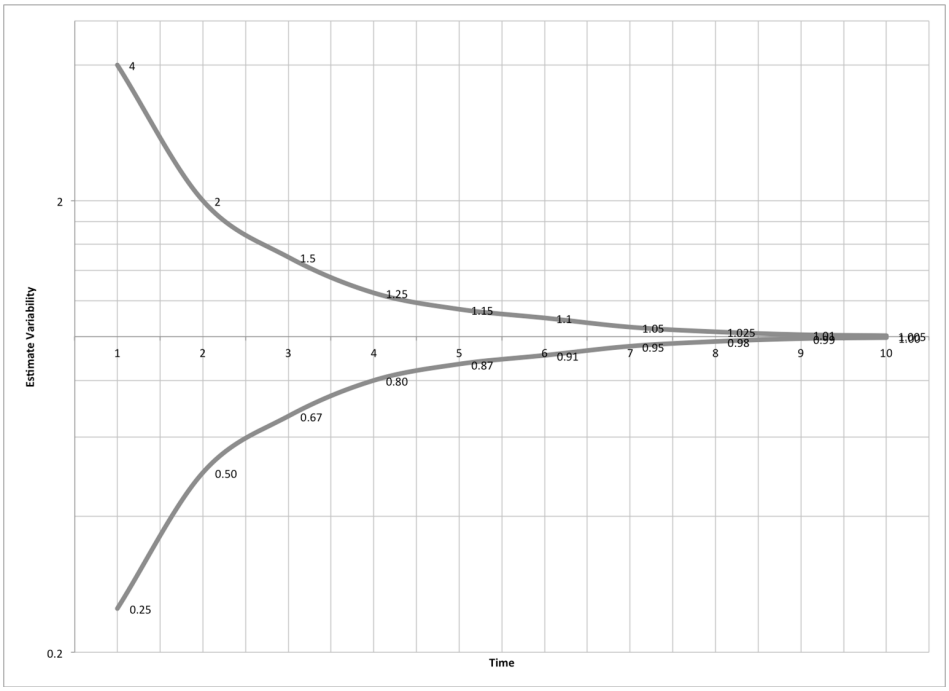
Apple Brook Consulting

**"The more precise the measurement of position,
the more imprecise the measurement of momentum,
and vice versa."**

-Werner Heisenberg
The Uncertainty Principle

**Estimation is ALWAYS inaccurate and imprecise.
We are trying to guess what will happen in the future.**

Cone of Uncertainty



The further out the planning horizon, the more variance there is in our estimates; i.e. when we estimate things that are further out, our estimates become less and less accurate.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

T-Shirt Sizes, Story Points, and NO Estimates

T-Shirt Sizes
XS, S, M, L, XL

(or, cute things like:
Cat sizes, fruit sizes, D&D creatures, planets, etc.)

Problem:

You can't add M+L+M+XS+M+XL to get a total size of a release

Story Points

Represent the concept of relative estimating
but in a mathematically quantifiable way

Integers:	**1, 2, 3, 4, 5**
Fibonacci *(modified)***:**	~~0, 1~~**, 1, 2, 3, 5, 8,** ~~13, 21…~~
Powers of 2:	**0, 1, 2, 4, 8, 16, 32…**

THERE IS NO CORRELATION BETWEEN POINTS AND TIME AT THE PBI LEVEL!!

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

"NO Estimates"

Many items, relatively equal size.
We track the count or number of items instead of points or hours.

"Does the complexity of this PBI require
more than a day of effort?"

YES: Split into smaller PBIs until each is
less than a day.

NO: The PBI is good to go.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Planning Poker

Planning Poker is a broad-band Delphi method by which Developers estimate Backlog Items using cards with a numeric sequence on them. The technique can be used for Product Backlog Items or Sprint Backlog Items including Tasks.

Various number sequences can be used including but not limited to Fibonacci, Powers of Two, Counting Numbers, T-Shirt Sizes, or whatever series the team agrees on. Once the series is selected, it should be kept consistent throughout the life of the Product to avoid the need for re-estimating.

Each person has a set and uses the set to vote on the relative size of the item. If there are disagreements, these are discussed and then votes are cast again. When there is consensus, the team moves on to the next item.



Typical Set of Planning Poker Cards

Planning Poker is an effective method for ensuring there is discussion amongst all Developers regarding the estimates. However, it can also be VERY time consuming.

Consider 30 items that must be estimated. If it takes 5-10 minutes / item to vote, discuss, re-vote and somehow come to consensus, it could easily take 2.5-5 hours just to estimate those items.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Affinity Estimating

Affinity Estimating is used when there are a large number of items to be estimated in a short space of time without engaging in lengthy discussions on each item. A number sequence is posted on a wall using the entire length of the wall and the numbers spread apart.

Items are then posted on the wall at roughly the size that the poster feels the item is. Others in the group review each item and move accordingly, taking into consideration items which are adjacent and in close proximity to each item. Traditionally, this is a silent exercise to ensure that it moves quickly.

At the end of the timeboxed period, the items are considered estimated and the value for the heading under which the item appears is assigned to the item. E.g. if the item is under the "5" it's estimated value is 5.

This method can be used to get estimates for a large collection of items in a very short period of time. The items will be discussed in more detail as time goes on. So the rough order of magnitude estimate is perfectly fine for the purposes of release planning.



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

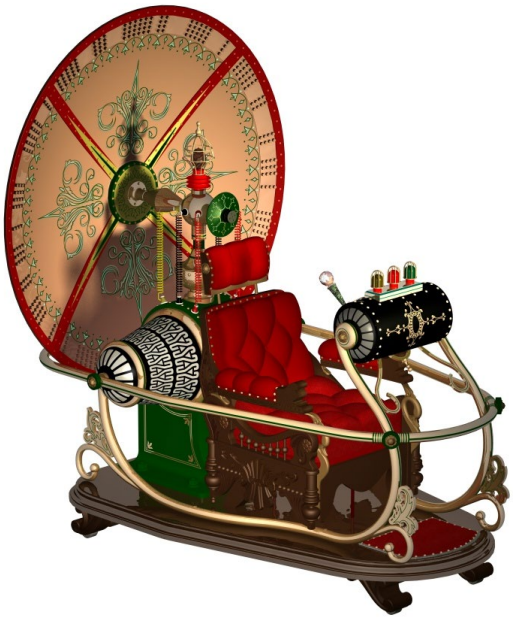
Real World Estimating Tools









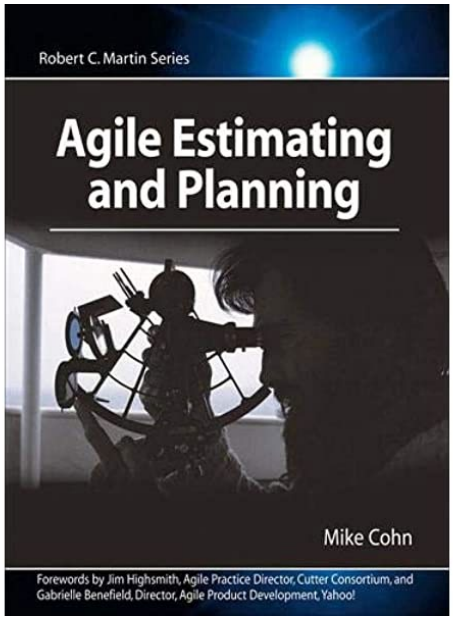




 Copyright © 2012-2025 - Daniel Gullo - All Rig

But seriously..

To learn more about Agile Estimating and Planning, read:





 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Velocity

Velocity is an average measure of work completed over a series of iterations; which is used to guide the Scrum Team during Release Planning.

Velocity is neither a measure of Project Success nor Team Performance.
Velocity is neither a measure of Project Success nor Team Performance.
Velocity is neither a measure of Project Success nor Team Performance.
(repeat)

Velocity does not replace collaborative decision making by the team in terms of what's possible for a Sprint.

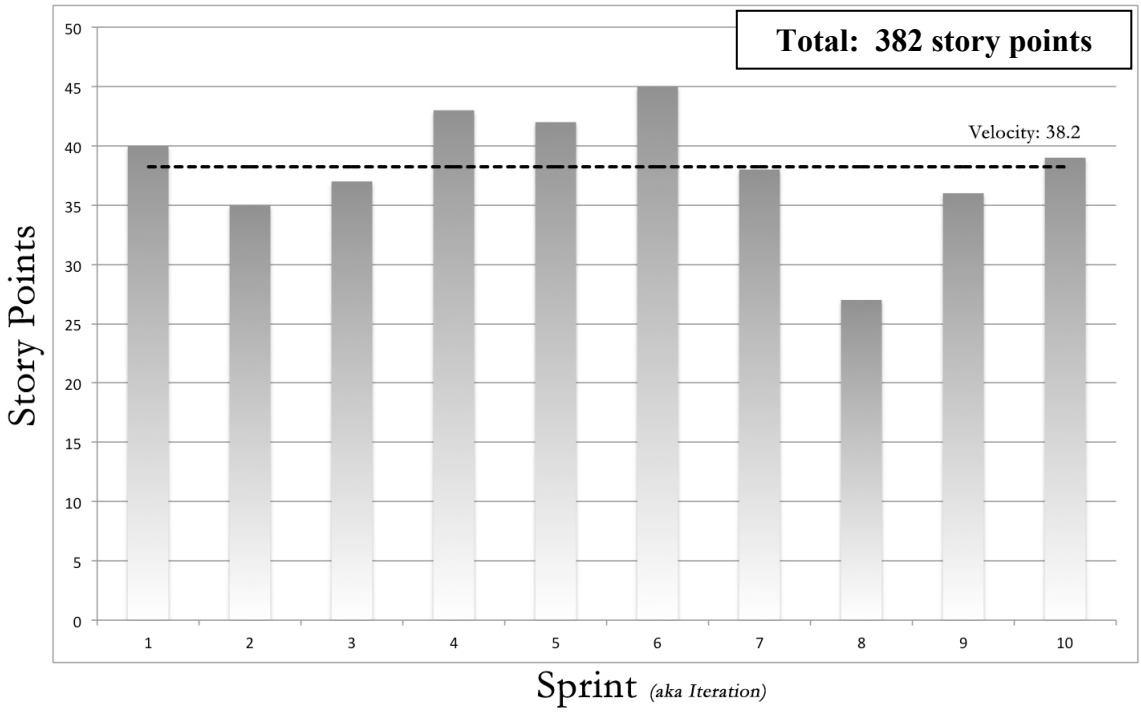
Together, the Developers and Product Owner decide what the Sprint Goal will be for the Sprint and will forecast what they can commit to deliver for the Sprint.

Velocity is one of the most misunderstood aspects of Scrum and is directly tied to Estimation, which is always inaccurate.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Velocity Chart



There are various philosophies around calculating the average. Some believe using the last three Sprints is more appropriate than the entire collection of known actuals for Sprints. Some do not use Sprints where there was less capacity due to PTO (vacations, people out sick, holidays, etc.)

Obviously, if the Velocity for the Developers is say 38 for a full Sprint, they would do less in a Sprint where there are vacations, etc.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Velocity Calculations

Fixed date – Today = # of days
of days / 7 = # of weeks
of weeks / Sprint duration = **# of Sprints**

Velocity x # of Sprints =
Estimated Scope

Example:
Developers that can do 40 items / Sprint on average must release by November 30. That date is 6 Sprints from today. How much scope (approximately) can the Developers complete?

Fixed Release Scope ÷ Velocity =
Estimated # of Sprints

Example:
A Product Owner indicates that the Release will happen when the MVP is ready, which is represented by 240 Story Points worth of Product Backlog Items. The Developers' velocity is 60 Story Points / Sprint. What is the estimated delivery date if today is February 1st our Sprints are 3 weeks in duration?

of Sprints x Sprint duration = # of weeks
of weeks x 7 = # of days
Today + # of days = **delivery date (estimated)**

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Sprint Burndown Examples

Figure 1

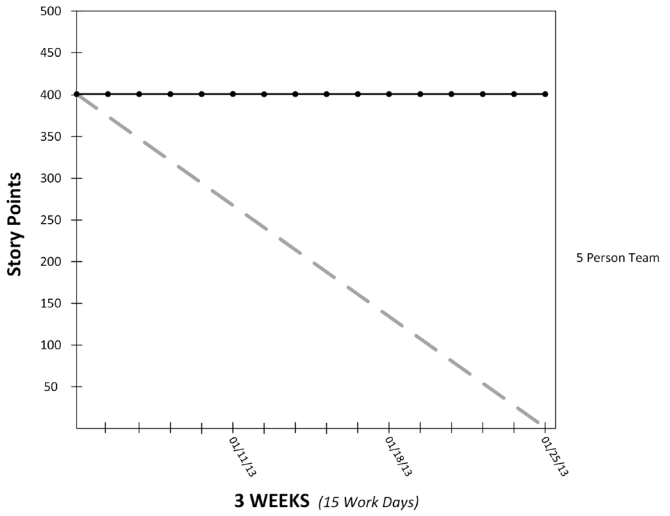


Figure 2

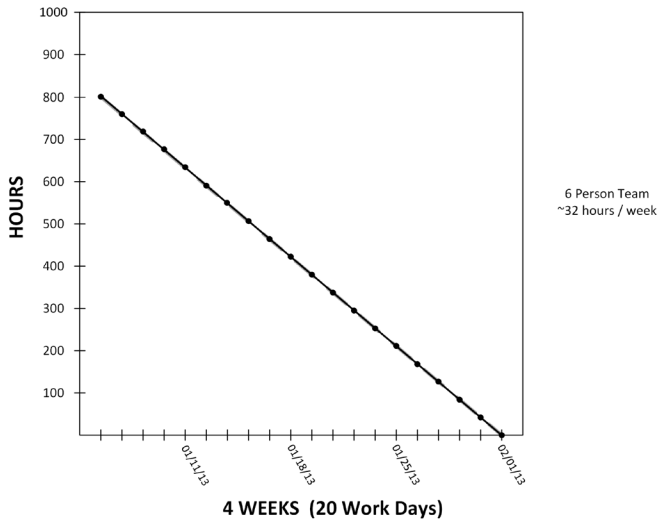
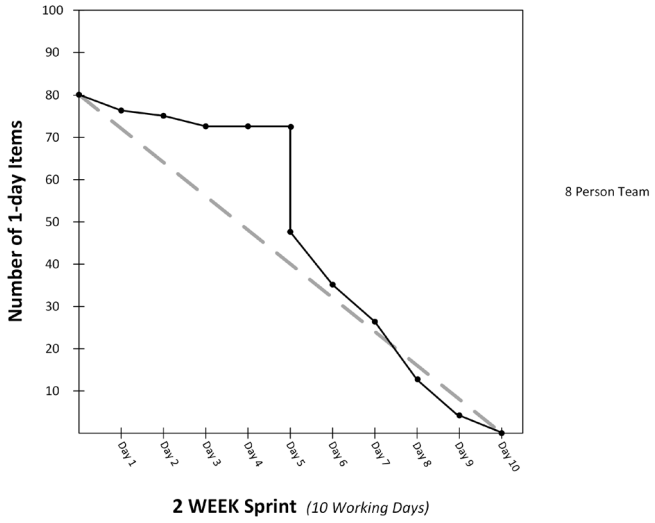


Figure 3

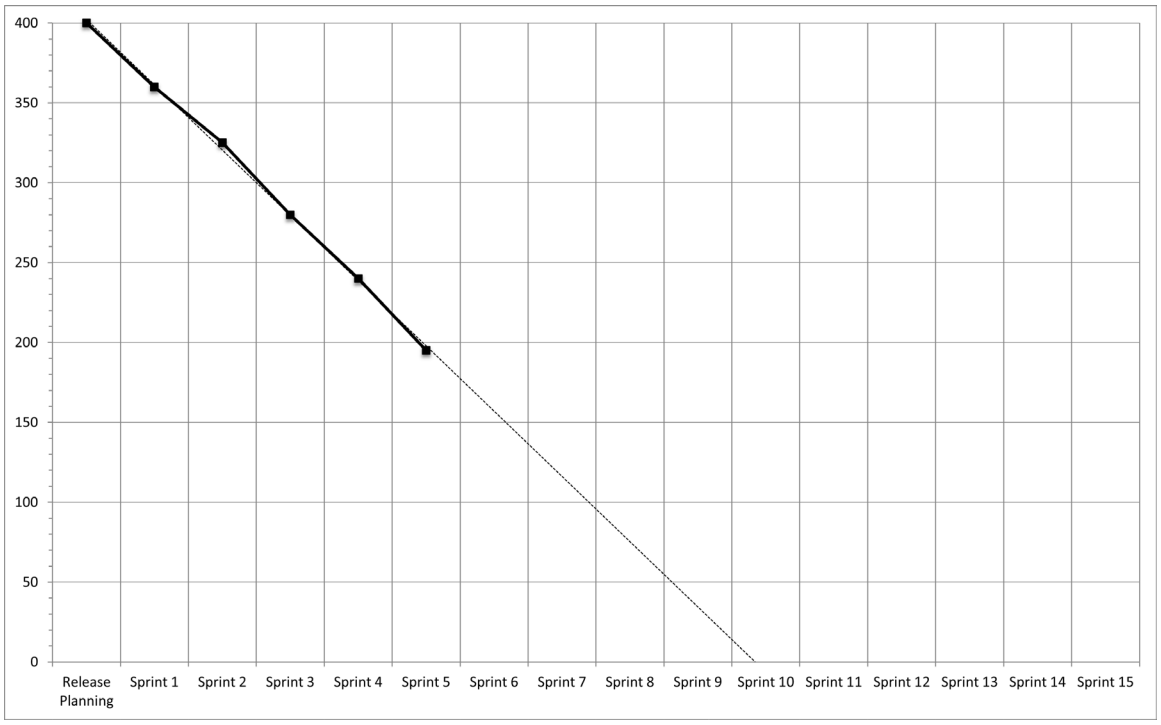


Sprint Burndowns should be updated at least DAILY.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Release Burndown Example



In this example, the team starts with 400 Story Points of scope identified (Product Backlog Items). The Developers begin working on these items and based on an average Velocity of 41 Story Points / Sprint, they will be done in 10 Sprints.

Release Burndowns should be updated after every Sprint.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

The **REAL** World

In the "real world," not everyone gets a trophy.

In the "real world," the lion EATS the gazelle.

In the "real world," if you DON'T learn how to inspect and adapt, then you are toast...

In the "real world," you have > 100 years of stock market data but you can't predict what a particular stock is going to do tomorrow, let alone next week, next month, 6 months from now, next year...

Like it or not, THAT is the "real world".

-Daniel Gullo, author: *Real World Agility*



 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Apple Brook Consulting

Bibliography

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

References and Further Reading

The books listed have had a deep and profound impact on my growth over the years as a coach, trainer, business owner, etc. I have drawn heavily upon these and others. I have done my best to provide a comprehensive list. If you get finished with these, there are hundreds of other books to continue on in your learning journey.

Adkins, Lyssa. 2010. *Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition.* Upper Saddle River, NJ: Addison-Wesley.

Anderson, David J. *Kanban: 2010. Successful Evolutionary Change for Your Technology Business.* Seattle: Blue Hole Press

Appelo, Jurgen. 2011. *Management 3.0: Leading Agile Developers, Developing Agile Leaders*. Upper Saddle River, NJ: Addison-Wesley.

Axelrod, Alan. 1999. *Patton On Leadership: Strategic Lessons for Corporate Warfare*. Paramus, NJ: Prentice Hall Press

Brooks, Frederick P. 1995. *The Mythical Man Month*. Reading, Mass: Addison-Wesley.

Buckingham, Marcus and Coffman, Curt. 1999. *First, Break All the Rules: What The World's Greatest Managers Do Differently*. New York: Simon and Schuster.

Csikszentmihalhi, M. (2020). Finding Flow: The Psychology Of Engagement With Everyday Life. United States: Basic Books

Cockburn, Alistair. 2000. *Writing Effective Use Cases*. New Jersey: Addison-Wesley.

Cohn, Mike. 2004. *User Stories Applied: For Agile Software Development.* Boston: Addison-Wesley.

Cohn, Mike. *2006. Agile Estimating and Planning.* Upper Saddle River NJ: Prentice Hall PTR.

Coplien, James O. 2005. *Organizational Patterns of Agile Software Development.* Upper Saddle River, NJ: Prentice Hall.

Denning, Stephen. 2010. *The Leader's Guide to Radical Management: Reinventing the Workplace for the 21st Century*. San Francisco: Jossey-Bass.

Derby, Esther and Larsen, Diana. 2006. *Agile Retrospectives: Making Good Teams Great.* Frisco, TX: Pragmatic Bookshelf

Foer, Joshua. 2011. *Moonwalking with Einstein: The Art and Science of Remembering Everything*. London: Penguin Books, Ltd.

Goggins, D. (2020). Can't Hurt Me: Master Your Mind and Defy the Odds - Clean Edition. United States: Lioncrest Publishing.

Goleman, Daniel. 2006. *Emotional Intelligence: Why It Can Matter More Than IQ*. New York: Random House

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Greenleaf, Robert K. 2002. *Servant Leadership: A Journey into The Nature of Legitimate Power and Greatness*. New Jersey: Paulist Press.

Gullo, Daniel J. 2016. *Real World Agility: Practical Guidance for Agile Practitioners.* Boston: Pearson Technical Group.

Highsmith, Jim. 2010. *Agile Project Management: Creating Innovative Products*. Upper Saddle River, NJ: Addison-Wesley.

Institute, P. M. (2017). Agile Practice Guide. United States: Project Management Institute.

Jeffries, Ron and Ann Anderson, Chet Hendrickson. 2000. *eXtreme Programming Installed*. Boston: Addison-Wesley.

Joiner, Bill and Stephen Josephs. 2007. *Leadership Agility: Five Levels of Mastery for Anticipating and Initiating Change*. San Francisco: Jossey-Bass.

Kerth, Norman. 2001. *Project Retrospectives: A Handbook for Team Reviews.* New York: Dorsett House

Kleon, Austin. 2012. *Steal Like an Artist: 10 Things Nobody Told You About Being Creative.* New York: Workman Publishing Company Inc.

Krebs, Jochen. 2009. *Agile Portfolio Management*. Redmond, WA: Microsoft Press.

Kohn, Alfie. 1999. *Punished by Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes*. Boston: Houghton Mifflin.

Kurzweil, Ray. 2012. *How to Create a Mind: The Secret of Human Thought Revealed.* New York: Penguin Books Ltd.

Larman, Craig. 2009. *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Upper Saddle River, NJ: Addison-Wesley.

Lasater, Judith Hanson and Lasater, Ike K. 2009. *What we Say Matters: Practicing Non-Violent Communication.* Berkeley: Rodmell Press.

Lencioni, Patrick. 2002. *The Five Dysfunctions of a Team: A Leadership Fable.* San Francisco: Jossey-Bass.

Leffingwell, Dean. 2011. *Agile Software Requirements: Lean Requirements Practices For Teams, Programs, and the Enterprise.* Upper Saddle River, NJ: Addison-Wesley.

Martinez, Sylvia Libow and Stager, Gary S. 2013. *Invent To Learn: Making, Tinkering, and Engineering in the Classroom*. Torrance, CA: Constructing Modern Knowledge Press.

McCarthy, Jim. 2002. *Software for Your Head: Core Protocols for Creating and Maintaining Shared Vision.* Boston: Addison-Wesley.

Rev: 11/07/2025 11:28:00 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

McConnell, Steven C. 2004. *Code Complete: A Practical Handbook Of Software Construction*. Redmond, WA: Microsoft Press.

Meadows, Donella H. 2008. *Thinking in Systems: A Primer.* Vermont: Chelsea Green Publishing Company

Mezick, Daniel. 2012. *The Culture Game: Tools for the Agile Manager*. Hartford: Free Standing Press.

Moore, Geoffrey A. 2002. *Crossing The Chasm*. New York: HarperCollins Publishers.

Owen, Harrison. 2008. *Open Space Technology: A User's Guide*. San Francisco: Berrett-Koehler Publishers, Inc.

Perkins, Dennis N. T. 2000. *Leading at the Edge : Leadership Lessons from the Extraordinary Saga of Shackleton's Antarctic Expedition.* New York: AMACOM

Pichler, Roman. 2010. *Agile Product Management with Scrum: Creating Products that Customers Love*. Upper Saddle River, NJ: Addison-Wesley.

Pink, Daniel. 2011. *Drive: The Surprising Truth About What Motivates Us. New York: Penguin Group.*

Poppendieck, M., & Poppendieck, T. (2003). *Lean software development: an agile toolkit*. Addison-Wesley.

Reinertsen, Donald G. 2009. *The Principles of Product Development FLOW: Second Generation Lean Product Development*. Redondo Beach, CA: Celeritas Publishing

Ries, Eric. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.* New York: Crown Business.

Robinson, Sir Ken. 2001. *Out of Our Minds: Learning to Be Creative*. West Sussex, United Kingdom: Capstone Publishing Ltd.

Rock, David and Linda J. Page. 2009. *Coaching with the Brain in Mind: Foundations for Practice.* Hoboken, NJ: John Wiley & Sons, Inc.

Rosenberg, Marshall. *2003. Non-Violent Communication: A Language of Life. Encinitas*, CA: PuddleDancer Press.

Satir, Virginia and Banmen, John and Gerber, Jane and Gomori, Maria. 1991. *The Satir Model: Family Therapy and Beyond*. Palo Alto, CA: Science and Behavior Books Inc.

Schwaber, Ken and Mike Beedle. 2002. *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice Hall.

Scott, Susan. 2011. *Fierce Conversations: Achieving Success at Work and in Life, One Conversation at a Time*. New York: The Berkley Publishing Group.

Seddon, John. 2005. *Freedom from Command and Control: Rethinking Management for Lean Service*. New York: Productivity Press.

Senge, Peter. 2006. *The Fifth Discipline*. New York: Currency – Doubleday.

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Shalloway, Alan and Beaver, Guy and Trott, James R. 2010. *Lean-Agile Software Development: Achieving Enterprise Agility*. Upper Saddle River, NJ: Addison-Wesley.

Sheridan, Richard. 2013. *Joy, Inc. How We Built a Workplace People Love*. New York: Penguin Group

Silsbee, Doug. 2008. *Presence-Based Coaching: Cultivating Self-Generative Leaders Through Mind, Body, and Heart.* San Francisco: Jossey-Bass.

Sinek, Simon. 2009. *Start with Why: How Great Leaders Inspire Everyone to Take Action*. New York: Penguin Group.

Sliger, Michele and Broderick, Stacia. 2008. *The Software Project Manager's Bridge to Agility*. Boston: Addison-Wesley.

Sutherland, Jeff. 2014. *Scrum: The Art of Doing Twice the Work in Half the Time*. New York: Crown Business.

Taleb, Nassim Nicholas. 2012. *Antifragile: Things That Gain from Disorder.* New York: Random House Publishing Group.

Weinberg, Gerald. 1998. *The Psychology of Computer Programming.* New York: Dorset House Publishing.

Weinberg, Gerald and Satir, Virginia. 1986. *The Secrets of Consulting.* New York: Dorset House Publishing

Wheatley, Margaret. 2006. *Leadership and the New Science: Discovering Order in a Chaotic World*. Oakland, CA: Berrett-Koehler Publishers Inc.

Willink, Jocko and Babin, Leif. 2015. *Extreme Ownership: How U.S. Navy Seals Lead and Win*. New York: St. Martin's Press

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved

Culturama

Various TV, Movie, and Music References

Being a visual and auditory person (and someone who enjoys laughing), I like to use a lot of imagery and analogies from movies, TV, music, etc. I don't make all of these references EVERY class, but in case you heard me reference something and were wondering "Huh?? What did he say?" here are some rough visuals that help me remember Agile concepts.

Star Wars: Episode IV – Being transparent vs. pretending everything is ok. A commander in the Death Star calls down to the detention block where Luke and Han have just taken control. Han says "We're all fine down here… how are you?"
Brazil – This whole movie is a classic parody of futuristic corporate life and bureaucracy. In particular, there is Harry Tuttle, the renegade HVAC repair man who says he left the Ministry because he couldn't stand the paperwork. "This whole place could be on fire and I couldn't turn a screwdriver without a 27B/6." This is the classic "comprehensive documentation" and "process and tools".
The Office (US) – Trusting technology too much and mindlessly following directions can have catastrophic consequences. Michael and Dwight found this out when Michael followed the GPS without using really thinking and drove right into a pond.
Spinal Tap – There is no "Sprint 0". Just like, it doesn't make sense to have amps that "go to 11".
Army of Darkness
Office Space
Columbo – Just one more question…
Poirot – The little grey cells are working overtime.
Mosquito Coast – In trying to escape technology and the trappings of modern culture, they bring these things with them and prove that man is inherently evil.
Trading Places – "Just be yourself, sir. No matter what happens, they can't take that away from you."

And many others…

 Copyright © 2012-2025 - Daniel Gullo - All Rights Reserved